

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

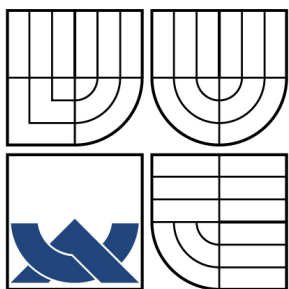
VÝBĚR VHODNÉHO PRAHOVACÍHO ALGORITMU PRO DETEKTOR
SRDEČNÍCH BUNĚK

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

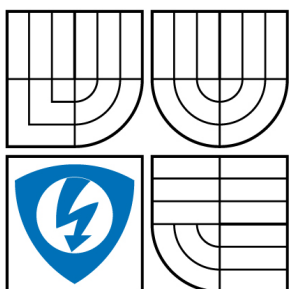
AUTOR PRÁCE
AUTHOR

MARTIN KOJAN

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

VÝBĚR VHODNÉHO PRAHOVACÍHO ALGORITMU PRO DETEKTOR SRDEČNÍCH BUNĚK

CHOOSING OF OPTIMAL TRESHOLD ALGORITHM FOR CARDIAC CELLS DETECTOR

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

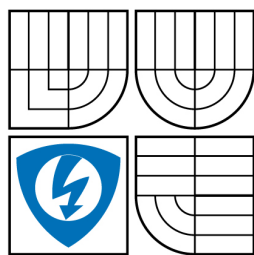
AUTOR PRÁCE
AUTHOR

MARTIN KOJAN

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MILAN RYCHTÁŘIK

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav biomedicínského inženýrství

Bakalářská práce

bakalářský studijní obor
Biomedicínská technika a bioinformatika

Student: Martin Kojan

ID: 22422

Ročník: 3

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Výběr vhodného prahovacího algoritmu pro detektor srdečních buněk

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte problematiku detekce hran v obraze a vyhodnocování kvality prahovacích metod. Vyberte vhodné prahovací algoritmy pro zpracování mikroskopických obrazů izolovaných srdečních buněk. Vybrané prahovací algoritmy realizujte v prostředí Matlab a realizované metody srovnajte. S ohledem na dané výsledky pak vyberte optimální prahovací algoritmy pro dodaný detektor pohybu srdečních buněk a tyto algoritmy realizujte v programu LabView jako nové moduly pro daný detektor. Algoritmy otestujte a zhodnotte především s ohledem na rychlost zpracování dat a kvalitu výsledné detekce.

DOPORUČENÁ LITERATURA:

- [1] Sezgin M, Sankur B: Survey over image thresholding techniques and quantitative performance evaluation, Journal of Electronic Imaging 13(1), 146–165 (January 2004).
- [2] Jan, J: Medical Image Processing, Reconstruction and Restoration. ISBN 0-8247-5849-8

Termín zadání: 8.2.2010

Termín odevzdání: 31.5.2010

Vedoucí práce: Ing. Milan Rychtárik

prof. Ing. Ivo Provazník, Ph.D.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cieľom tejto práce je realizácia detektoru kontrakcie srdcových buniek v programe LabView. Zameriava sa na problematiku prahovania obrazu, návrh a realizáciu prahovacích algoritmov a ich následnú implementáciu do detektora.

KLÚČOVÉ SLOVÁ

srdcové bunky, detektor, kontrakcia, prahovanie, prahovací algoritmus, spracovanie obrazu

ABSTRACT

The objective of the thesis is the realisation of detector for cardiac cell's contraction in LabView software. The thesis is focused on the issue of image thresholding, suggestion and realisation of threshold algorithms and the following implementation into the detector.

KEYWORDS

cardiac cells, detector, contraction, thresholding, thresholding algorithm, image processing

KOJAN, Martin *Výběr vhodného prahovacího algoritmu pro detektor srdečních buněk*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2010. VI s. Vedoucí práce byl Ing. Milan Rychtářík

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Výběr vhodného prahovacího algoritmu pro detektor srdečních buněk“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno

.....

(podpis autora)

Ďakujem vedúcemu bakalárskej práce Ing. Milanovi Rychtárikovi za účinnú metodickú, pedagogickú a odbornú pomoc a ďalšie cenné rady pri spracovávaní záverečnej práce. Zároveň ďakujem aj Ing. Vratislavovi Čmielovi za jeho odbornú pomoc pri realizácii detektoru v prostredí LabVIEW.

OBSAH

1	Úvod	5
2	Analyzované obrazy	6
2.1	Optická mikroskopia	6
2.2	Kardiomyocyty	6
3	Problematika spracovania obrazu	8
3.1	Spracovanie farebného obrazu	8
3.2	Spracovanie obrazu v odtieňoch šedej	8
4	Prahovanie obrazu	9
4.1	Histogram obrazu	9
4.2	Globálne prahovanie	10
4.3	Prahovanie s hysteréziou	11
4.4	Adaptívne prahovanie	11
4.5	Metódy hľadania vhodného prahu	12
4.5.1	Metóda Otsu	12
4.5.2	Chow a Kanenko metóda	12
4.5.3	Lokálne prahovanie	12
4.6	Ďalšie možnosti a metódy prahovania	12
4.6.1	Metódy využívajúce tvar histogramu	13
4.6.2	Metódy založené na zhlukovaní	13
4.6.3	Metódy využívajúce entropiu obrazu	14
4.6.4	Metódy hodnotiace atribúty obrazu	14
4.6.5	Priestorové prahovacie metódy	14
4.6.6	Lokálne adaptívne prahovanie	15
5	Realizácia prahovania v prostredí Matlab	16
5.1	Prahovanie s využitím mediánového filtra	17
5.2	Prahovanie pomocou priemerovacieho filtra	17
5.3	Metóda Otsu	17
5.4	Prahovanie s hysteréziou	18
5.5	Iteračné prahovanie	19
5.6	Prahovanie s využitím informácie o entropii obrazu	19
6	Porovnanie výsledkov	22
6.1	Porovnanie výpočtovej náročnosti jednotlivých metód	22
6.2	Subjektívne hodnotenie výsledkov prahovania	23

6.3	Objektívne kritériá pre hodnotenie prahovania	23
6.4	Vyhodnotenie kvality vybraných prahovacích metód pomocou objek- tívnych kritérií	25
7	Prahovanie obrazu v LabVIEW	28
7.1	Implementácia prahovacích algoritmov do LabVIEW	28
7.1.1	Možnosti realizácie	28
7.1.2	Popis bloku Thresholding.vi	29
7.1.3	Program prahovanie.vi	29
7.2	Detektor kontrakcie srdcových buniek	30
7.2.1	Realizácia detektoru	32
7.2.2	Analýza krivky kontrakcie	34
8	Záver	35
	Literatúra	36
	Zoznam príloh	37
A	Program prahovanie.vi	I
A.1	Návod k obsluhu	I
A.2	Galéria obrázkov	II
B	Návod na obsluhu detektoru	IV
B.1	Ukážky funkcií detektoru	V

ZOZNAM OBRÁZKOV

2.1	Kardiomyocyt na snímke z optického mikroskopu Olympus IX71 (zväčšené 200x)	7
4.1	Histogram a ekvalizovaný histogram obrazu	9
4.2	Pôvodný a ekvalizovaný obraz	10
5.1	Izolované kardiomyocyty z potkana zobrazené s využitím metódy fázového kontrastu	16
5.2	Prahovanie s využitím filtrov	18
5.3	Prahovanie metódou Otsu	18
5.4	Hysterézne prahovanie	18
5.5	Obraz po iteračnom prahovaní	19
5.6	Prahovanie s využitím informácie o entropii obrazu	20
5.7	Obraz 5.6 filtrovaný mediánovým filtrom s veľkosťou okna 3x3	21
6.1	Výrez z Obr. 5.1 (vľavo) a jeho ideálny binárny obraz(vpravo)	26
6.2	Výsledok prahovania výrezu z Obr. 6.4 pomocou metódy Otsu	26
6.3	Pomocné obrazy určené k výpočtu hodnoty ME	27
7.1	Bloková schéma realizácie prahovacích algoritmov	30
7.2	Okno programu pre prahovanie statického obrazu	30
7.3	Realizácia pracoviska pre detekciu kontrakcie srdcových buniek	31
7.4	Návrh systému pre detekciu kardiomyocytov	32
7.5	Okno programu zobrazujúce krivku kontrakcie	33
A.1	Dialógové okno pre výber oblasti záujmu	II
A.2	Prahovanie vybranej oblasti záujmu	II
A.3	Príklad prahovania naskenovaného textu	III
B.1	Záložka detektoru s možnosťami spracovania obrazu	V
B.2	Karta programu s vyobrazenou krivkou kontrakcie	V
B.3	Histogram a rozlíšenie analyzovaného obrazu	VI
B.4	Detail boxu pre nastavenie snímacích parametrov	VI
B.5	Detail boxu s prepočítanými dĺžkami jednotlivých častí krivky	VI

ZOZNAM TABULIEK

6.1	Časy potrebné na prahovanie	22
6.2	Hodnoty ME pre jednotlivé prahovacie metódy	27

1 ÚVOD

Táto práca má za úlohu vyhľadať vhodnú prahovaciu metódu pre detekciu kontrakcie srdcových buniek na snímkoch zhotovených pomocou kamery pripojenej k optickému mikroskopu. V úvodných kapitolách tejto práce sa oboznámime s problematikou teórie spracovania obrazu, konkrétne prahovaním obrazu. Nachádza sa tu základné rozdelenie a prehľad existujúcich metód prahovania obrazu. V praktickej časti sú vybrané metódy najskôr realizované v programovom prostredí Matlab. Následne sú ich výsledky zhodnotené a porovnané s ohľadom na kvalitu a hlavne rýchlosť spracovania. V ďalšej časti sa táto práca venuje realizácii a zapracovaniu metód, ktoré spĺňajú požiadavky na kvalitu a rýchlosť spracovania do programu LabVIEW. V prvom kroku sú implementované do programu na prahovanie statického obrazu, kde je overená ich funkčnosť a rýchlosť. V záverečnej časti práce je popísaná realizácia detektoru kontrakcie, jeho využitie v praxi a popis jeho základných častí, vstupov a výstupov. V prílohe práce sa nachádzajú jednoduché návody na obsluhu realizovaných programov a ich stručná fotodokumentácia.

2 ANALYZOVANÉ OBRAZY

Objekty, ktoré nás budú zaujímať pri analýze, spracovaní a prahovaní obrazu sú izolované srdcové bunky – kardiomyocyty. Obrázky kardiomyocytov boli zhotovené pomocou vysokorýchlostnej kamery Olympus iSpeed3. Rozlíšenie jednotlivých snímkov je 640x512 bodov pri 400-násobnom zväčšení.

2.1 Optická mikroskopia

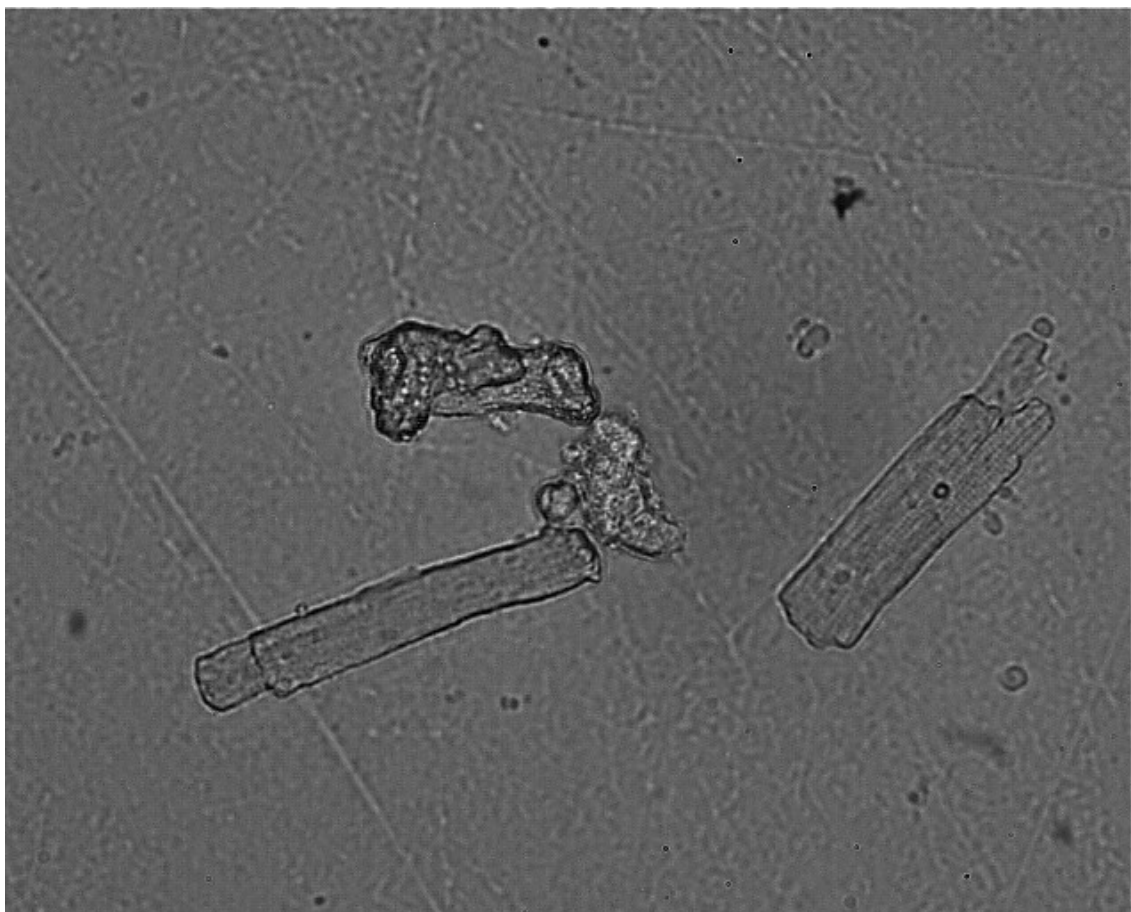
Zdroj obrazových údajov, ktoré sú analyzované v rámci tejto práce je optický mikroskop. Medzi základné časti optického mikroskopu patria: okulár, objektív a kondenzor. Primárny obraz je tvorený objektívom, ďalej je obraz zväčšený pomocou okuláru za vzniku výsledného obrazu. Celková optická mohutnosť mikroskopu sa spočíta ako súčin optických mohutností objektívu a okuláru. Pre pozorovanie a skúmanie kardiomyocytov sa používa zväčšenie 400x. K takejto optickej sústave je pripojená vysokorýchlostná kamera Olympus iSpeed3, ktorá zaznamenáva jednotlivé obrazy kardiomyocytov v digitálnej podobe.

2.2 Kardiomyocyty

Svalové tkanivo, ktoré vytvára srdcový sval je svojou stavbou a štruktúrou veľmi podobné priečne pruhovanému kostrovému svalstvu. Ich jadrá sú ale umiestnené uprostred buniek a vytvárajú tak priestor pre mikrofilamenty kontraktilného aparátu. [1] Z hľadiska funkcie môžeme rozdeliť kardiomyocyty na pracovné a vzrušivé. Pracovné (kontraktilné) kardiomyocyty tvoria strednú vrstvu srdcovej svaloviny (myokard). Sú to bunky valcovitého tvaru s dĺžkou 85 – 100 μm a šírkou 10 – 15 μm . Kontraktilné kardiomyocyty sa svojimi koncami zoraďujú lineárne do tenkých paralelných vlákien, ktoré špirálovo prebiehajú od bázy k srdcovému hrotu. Rozhrania medzi jednotlivými bunkami sa javia ako tmavšie pruhy označované ako interkalárne disky. Kardiomyocyty obsahujú veľké množstvo mitochondrií a to preto, že musí byť zabezpečená nepretržitá práca po celú dobu života. Ku kontrakcii myokardu sa využíva jedine aeróbny metabolizmus, ktorý je regulovaný samotným srdcovým svalom (autonómny srdcový systém). Zdrojom energie pre myokard sú mastné kyseliny a pri hladovaní tiež ketolátky. Myokard na rozdiel od kostrového svalstva nemôže pracovať na kyslíkový dlh a vyžaduje teda stály prívod kyslíku. [1] Srdcové bunky nemajú schopnosť regenerácie a ak sú poškodené alebo zničené, sú nahradené väzivom (jazvy na myokarde).

Každá srdcová bunka má v určitej miere vyvinuté tieto štyri vlastnosti: [2]

- automaciu, je schopná rytmicity, tj. pravidelnej tvorby elektrických vzruchov
- vodivosť, elektrický impulz vzniknutý na ktoromkoľvek mieste myokardu je rozvedený po celom myokarde
- dráždivosť, po podráždení nadprahovým podnetom získame reakciu kardiomyocytov
- sťahovateľnosť, schopnosť kardiomyocytov reagovať na podráždenie svojim stiahnutím



Obr. 2.1: Kardiomyocyt na snímke z optického mikroskopu Olympus IX71 (zväčšené 200x)

3 PROBLEMATIKA SPRACOVANIA OBRAZU

3.1 Spracovanie farebného obrazu

Naším cieľom pri spracovávaní obrazu je prevod farebného výstupu na monochromatický, obsahujúci len čiernu a bielu farbu. V prvom kroku prevedieme farebný obraz na nový v odtieňoch šedej. Pre každý bod v obraze(pixel) platí, že jeho farbu môžeme vyjadriť ako vektor spektrálnych činiteľov troch základných farieb: červenej(R), zelenej(G) a modrej (B). [3]

$$C |r, g, b| = r.R + g.G + b.B \quad (3.1)$$

kde r, g, b su intenzity jas u uvedených farieb. Ľudské oko nevníma všetky farby rovnako, najcitlivejšie je na zelenú farbu. Keď chceme, aby oko vnímalo obraz len v odtieňoch šedej, musíme každú zložku vynásobiť určitými rozdielnymi koeficientami. [3] Výsledný vzorec prevodu farebného obrazu do obrazu šedotónového potom vyzerá nasledujúco:

$$I = 0,299.R + 0,587.G + 0,114.B \quad (3.2)$$

Vypočítanú intenzitu svetla I potom dosadíme za každú z farieb R, G, B.

3.2 Spracovanie obrazu v odtieňoch šedej

Po prepočítaní všetkých pixelov podľa vyššie uvedeného algoritmu sme získali obraz v odtieňoch šedej, ktorý už je vhodný pre ďalšie matematické operácie. V ďalšom kroku je našou úlohou pomocou vhodnej metódy získať obraz, ktorý bude obsahovať už len čiernu a bielu farbu. Budeme skúmať možnosti oddelenia objektov v popredí obrazu od jeho pozadia.

4 PRAHOVANIE OBRAZU

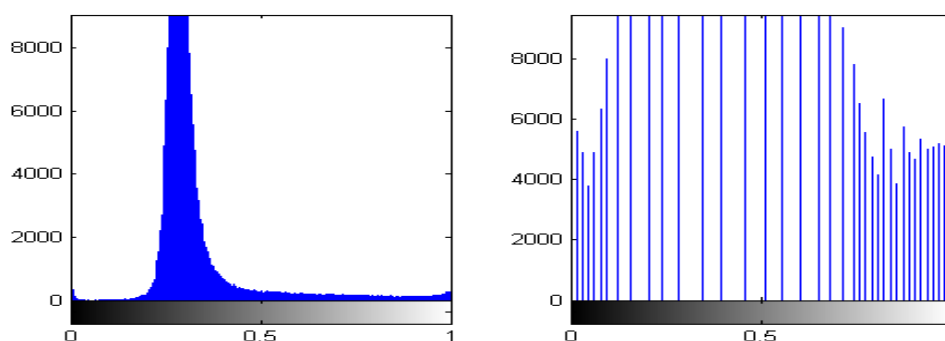
Metódy určené k rozdeleniu obrazu na oblasti, ktoré nás zaujímajú – popredie a oblasti bez užitočných a použiteľných údajov – pozadie, sa súhrnne označujú ako prahovanie obrazu. Ich spoločným cieľom je nájsť nejakú konkrétnu hodnotu odtieňa šedej – prah. Body, ktoré majú svetlejší odtieň ako je prah sú identifikované ako pozadie. Body tmavšie ako prah sú brané ako objekty v popredí. Výsledný obraz potom zobrazuje čierne objekty na bielom pozadí. Keď si označíme prah T a $f(x,y)$ je pôvodný obraz, tak pre výsledný obraz $b(x,y)$ platí:

$$b(x,y) = \begin{cases} 255, & \text{if } f(x,y) > T \\ 0, & \text{if } f(x,y) \leq T \end{cases} \quad (4.1)$$

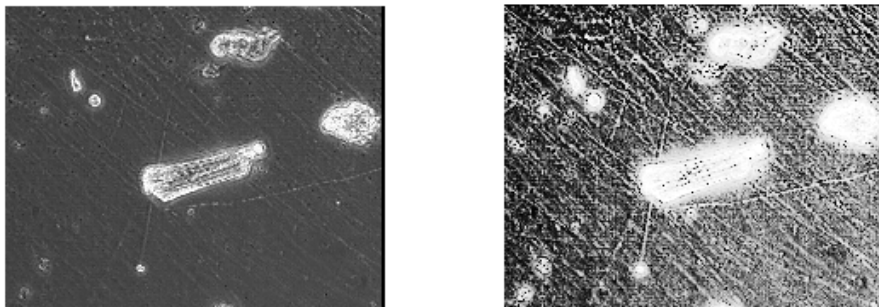
Celá problematika prahovania je teda orientovaná na nájdenie vhodného prahu T pre daný obraz. K tomuto účelu slúži celá rada metód a matematických postupov, ktorým sa budem venovať v nasledujúcom texte.

4.1 Histogram obrazu

Histogram jasu daného obrazu $f(x,y)$ je funkcia $H(p)$, ktorá pre každú úroveň jasu udáva počet pixelov, ktoré majú v obraze túto úroveň. Dá sa chápať aj ako kvantovaná hustota pravdepodobnosti. [4] Pre zvýšenie kontrastu v obraze sa často využíva ekvalizácia histogramu. Ekvalizáciou sa snažíme dosiahnuť ideálny histogram obsahujúci rovnaký počet pixelov z každej zastúpenej jasovej hodnoty. Na Obr. 4.1 vidíme histogram a ekvalizovaný histogram, na Obr. 4.2 je pôvodný obraz a obraz po ekvalizácii histogramu.



Obr. 4.1: Histogram a ekvalizovaný histogram obrazu



Obr. 4.2: Pôvodný a ekvalizovaný obraz

4.2 Globálne prahovanie

Najzákladnejšou prahovacou metódou je globálne prahovanie – global thresholding. Pri tomto postupe máme na vstupe obraz v odtieňoch šedej. Do procesu nám ďalej vstupuje jediný atribút a tým je pevná hodnota globálneho prahu. [3] Každý bod z obrazu je následne porovnávaný s touto hodnotou intenzity jasu a je mu podľa vzťahu (4.1) priradená nová hodnota. Otázkou ale zostáva, ako zvoliť hodnotu prahu T . K tomu nám pri tejto metóde posluží histogram skúmaného obrazu.

Ak histogram obrazovej funkcie je bimodálny, to znamená, že predmety v obraze majú výrazne odlišnú hodnotu jasu od pozadia. Lokálne minimum medzi dvoma špičkami histogramu je vhodné použiť ako prahovú hodnotu. [4] V prípade, že rozdelenie intenzít jasu nie je možné rozdeliť do dvoch skupín, môžeme predpokladať neuspokojivý výsledok prahovania a na takéto obrazy musíme použiť odlišné prahovacie postupy.

4.3 Prahovanie s hysteréziou

Táto metóda nám vnáša do procesu ďalší parameter δ – mieru hysterézie. Musíme ju stanoviť tak, aby bola splnená podmienka $T - \delta > 0$. S pomocou tohoto parametru potom určíme dve nové hodnoty. [5] $T_{HIGH} = T$ a $T_{LOW} = T - \delta$. Obrazová funkcia $b(x, y)$ bude potom nadobúdať nasledujúce hodnoty:

- Pre body s hladinou intenzity nižšou ako T_{LOW} sa priradí hodnota 0
- Body s hodnotou jasu vyššou ako T_{HIGH} majú priradenú novú hodnotu 1
- Pre body spadajúce do intervalu $< T_{LOW}, T_{HIGH} >$ je priradená hodnota 1 jedine v prípade, že v okolí bodu sa už nachádza iný bod s hodnotou 1, inak je mu priradená hodnota 0.

Okolím bodu sa rozumejú body, ktoré susedia s daným bodom v zmysle bežne používaných pojmov 4-susednosti a 8-susednosti v digitálnom obraze. [5]

4.4 Adaptívne prahovanie

V prípadoch, kedy obraz na vstupe obsahuje objekty, ktoré nie sú dostatočne osvetlené alebo vrhajú tieň, nie je možné pomocou globálneho prahovania dosiahnuť požadovaný výsledok. Pevný prah nie je schopný odlíšiť všetky objekty od pozadia. Preto musíme využiť adaptívne prahovanie, kde sa prahovacia hodnota vypočítava pre každý bod zvlášť. Definíciu prahovej hodnoty teda môžeme zapísať ako:

$$T = T[x, y, p(x, y), f(x, y)]$$

kde $p(x, y)$ je určitá lokálna vlastnosť bodu (x, y) a $f(x, y)$ je stupeň šedi daného bodu. Faktor $p(x, y)$ je dôležitý prvok pri výpočte prahovej hodnoty pre bod v obraze. Výpočet tohoto parametra je obvykle založený na okolí bodu. Na jeho vyjadrenie zvyčajne vychádzame z priemeru jasových hodnôt v okolí bodu, mediánu jasových hodnôt okolia alebo strednej hodnoty minimálnej a maximálnej jasovej hodnoty v okolí. Okolie musíme zvoliť tak, aby obsahovalo dostatočne veľký počet bodov objektu aj pozadia. Adaptívne prahovanie teda počíta hodnotu prahu pre každý bod poľa a dynamicky ju mení nad spracovávaným obrazom. Dochádza k lepšiemu prispôsobeniu sa rozdielnej intenzite osvetlenia a vzniknutým tieňom. [3]

Pre hľadanie prahovej hodnoty sa používa mnoho postupov, medzi najrozšírenejšie patrí metóda Otsu, algoritmus Chow-Kaneneko, lokálne prahovanie a iné. V nasledujúcich kapitolách sa s niektorými metódami zoznámime bližšie.

4.5 Metódy hľadania vhodného prahu

4.5.1 Metóda Otsu

Metóda Otsu je používaná pre prahovanie obrazu alebo pre jeho prevod zo šedotónového do binárneho. Algoritmus predpokladá existenciu dvoch tried pixelov (objekty a pozadie) a pomocou štatistických metód počíta optimálny prah, ktorý tieto dve triedy oddeľuje a medzitriedna variancia je najmenšia. Táto metóda je bližšie popísaná v kapitole 4.6.2

4.5.2 Chow a Kanenko metóda

Táto metóda vychádza z predpokladu, že presnejšiu hodnotu prahu môžeme získať z menšieho obrazu. Rozdeľuje teda vstupný obraz na množinu prekrývajúcich sa podobrazov a pre každý z nich určujeme z ich histogramu prahovú hodnotu. [4] Tá je pre každý pixel vypočítaná ako interpolácia daných podobrazov. [3] Metóda je náročná na čas a výpočtový výkon, preto je nevhodná pre aplikácie s požiadavkami na prácu v reálnom čase.

4.5.3 Lokálne prahovanie

Pri metóde lokálneho prahovania štatistickými metódami vyhodnocujeme pre každý pixel hodnotu intenzity jasu v jeho blízkom okolí. Funkcia, ktorú využívame a aj veľkosť okolia, ktoré berieme do úvahy, sú závislé od predloženého obrazu. Medzi najjednoduchšie a najrýchlejšie patrí medián a stredná hodnota. Pre zvolený prah T potom platí, že jeho veľkosť je rovná strednej hodnote intenzít jasu v okolí daného bodu, prípadne ich mediánu. Veľkosť zvoleného blízkeho okolia musí byť zvolená tak, aby pokryla dostatočnú časť objektu aj pozadia. [3] Zväčšovaním blízkeho okolia však rastú požiadavky algoritmu na výpočtový výkon a môže byť porušená podmienka homogenity osvetlenia.

Lokálne adaptívne prahovanie teda môžeme použiť na obrazy nevhodné pre globálne prahovanie, ktoré nemajú dostatočne homogénne osvetlenie a z ich histogramu sa nedá spoľahlivo určiť prahová hodnota.

4.6 Ďalšie možnosti a metódy prahovania

Prahovacie algoritmy môžeme podľa inej klasifikácie rozdeliť do ďalších šiestich kategórií. Konkrétne sa jedná o algoritmy založené na využívaní:

1. informácie o tvare histogramu
2. merania zhľukovania priestoru

3. informácie o entropii obrazu
4. informácie o atribútoch obrazu
5. priestorovou informáciu
6. lokálne charakteristiky

Pre každú kategóriu je charakteristický konkrétny prístup k hľadaniu optimálneho prahu a existuje viacero možností, ako je možné tento prah vypočítať. [6]

4.6.1 Metódy využívajúce tvar histogramu

Pri algoritmoch z tejto kategórie sa k analýze a výpočtu optimálneho prahu využíva tvar histogramu – jeho špičky, poklesy, zakrivenia. Niektoré metódy počítajú vzdialenosť od konvexnej obálky histogramu, iné sa snažia modifikovať histogram do dvoch peakov pomocou autoregresívneho modelovania. Ešte hrubší odhad je dosiahnutý pomocou pravouhlej aproximácie. [6] Niektoré algoritmy explicitne hľadajú funkčné maximá a minimá histogramu.

4.6.2 Metódy založené na zhlukovaní

Pre túto triedu algoritmov platí, že pre výpočet využívajú zhlukovú analýzu a počet zhlukov je pevne nastavený na dva. Tieto reprezentujú priamo dva laloky histogramu, pričom niektoré metódy hľadajú stred medzi dvoma peakmi. Ďalej môžeme použiť algoritmy na základe Gaussovských funkcií, zhlukovanie podľa strednej kvadratickej odchýlky a metódy založené na fuzzy logike. [6]

Do tejto kategórie patrí už aj vyššie spomenutá metóda Otsu (kap. 4.5.1). Táto navrhuje k nájdeniu optimálneho prahu minimalizáciu sumy medzitriednej variancie pixelov popredia a pozadia. Poskytuje uspokojivé výsledky v situáciách, kedy sú výpočty pixelov v oboch triedach porovnateľné. [6] Metóda Otsu stále zostáva jednou z najviac odkazovaných prahovacích metód. Pre výpočet optimálneho prahu využíva nasledujúci vzťah:

$$T_{opt} = \arg \max \left\{ \frac{P(T) |1 - P(T)| |m_f(T) - m_b(T)|}{P(T) \sigma_f^2(T) + |1 - P(T)| \sigma_b^2(T)} \right\} \quad (4.2)$$

Ďalšia z metód spadajúcich do tejto kategórie je iteračné prahovanie. Riddler [6] vylepšil prvé iteračné schémy založené na Gaussovských modeloch dvoch zmiešaných tried. Matematické vyjadrenie je založené na stanovení nového prahu pomocou aritmetických priemerov popredia a pozadia v každom kroku iterácie. Výpočet je ukončený pri dostatočne malom rozdieli medzi dvoma susednými krokmi. Vyjadríme to nasledovne:

$$T_{opt} = \frac{\lim_{n \rightarrow \infty} m_f(T_n) + m_b(T_n)}{2}, \quad (4.3)$$

$$m_f(T_n) = \sum_{g=0}^{T_n} gp(g)m_b(T_n) = \sum_{g=T_n+1}^G gp(g)$$

4.6.3 Metódy využívajúce entropiu obrazu

Táto trieda algoritmov využíva entropiu rozdelenia jednotlivých úrovní šedej v scéne. Maximalizácia entropie prahovaného obrazu je považovaná za znak maximálneho prenosu obrazovej informácie. Niektorí autori sa snažia minimalizovať krížovú entropiu medzi originálnym obrazom v odtieňoch šedej a výstupným binárnym obrazom pre maximálne zachovanie obrazovej informácie.

Pre praktickú časť som z tejto kategórie využil výpočet podľa Kapur, Sahoo a Wong. [6] Táto metóda chápe objekty a pozadie ako dva samostatné zdroje signálu. Obraz považuje za správne prahovaný v prípade, keď dosiahne krížový súčet entropií svojho maxima. [6] Výpočet optimálneho prahu potom vyjadruje vzťah (4.4):

$$T_{opt} = \operatorname{argmax} [H_f(T) + H_b(T)], \text{ kde} \quad (4.4)$$

$$H_f(T) = \sum_{g=0}^T \frac{p(g)}{P(T)} \log \frac{p(g)}{P(T)} \text{ a } H_b(T) = \sum_{g=T+1}^G \frac{p(g)}{P(T)} \log \frac{p(g)}{P(T)}$$

4.6.4 Metódy hodnotiace atribúty obrazu

Tieto algoritmy určujú hodnotu prahu na základe vlastností niektorých atribútov alebo merania podobnosti medzi pôvodným a binarizovaným obrazom. Tieto atribúty môžu byť založené na detekcii hrán, kompaktnosti tvaru, stabilite segmentovaných objektov, či konektivite. Prahovanie založené na detekcii hrán podľa Hertz a Schaffer [6] porovnáva pole hrán získané z pôvodného obrazu E_{gray} s hranami prahovaného obrazu $E_{binary}(T)$. Optimálny prah je stanovený ako hodnota maximálnej zhody týchto dvoch polí.

$$T_{opt} = [E_{gray} \cap E_{binary}(T)] \quad (4.5)$$

4.6.5 Priestorové prahovacie metódy

Táto kategória algoritmov využíva pre výpočet optimálneho prahu nielen rozdelenie hodnôt intenzity jasu, ale aj závislosti bodov na svojom okolí. Pravdepodobnosť výskytu, korelačné funkcie, pravdepodobnosť súčasného výskytu a lokálnu lineárnu závislosť. [6]

4.6.6 Lokálne adaptívne prahovanie

Ako už bolo spomenuté v predchádzajúcom texte, tento druh prahovacích metód počíta hodnotu prahu pre každý bod obrazu osobitne. Výpočet je závislý na niektorých lokálnych štatistických parametroch, akými sú rozptyl alebo odchyľka. Následne je prah $T(i, j)$ vyjadrený ako funkcia súradníc i, j . [3]

5 REALIZÁCIA PRAHOVANIA V PROSTREDÍ MATLAB

V úvode praktickej časti tejto práce boli realizované niektoré z prahovacích metód v programovom prostredí Matlab. Snímky použité k prevereniu jednotlivých algoritmov, boli nasnímané vysokorýchlostnou kamerou Olympus iSpeed3. Sú vo formáte .jpg, majú rozlíšenie 640x512 pixelov pri 400-násobnom zväčšení. Pre lepšiu identifikáciu buniek v obraze boli zhotovené metódou fázového kontrastu.



Obr. 5.1: Izolované kardiomyocyty z potkana zobrazené s využitím metódy fázového kontrastu

Základ realizovaného programu pre prahovanie tvorí skript, ktorý využíva základné funkcie jazyka Matlab pre prácu s obrazom. V tejto fáze je analyzovaný obraz načítaný, prevedený do šedotónového obrazu, vypočítaný jeho histogram a ekvalizovaný histogram. V nasledujúcich krokoch program využíva ku svojej činnosti samostatné skripty pre každý realizovaný druh prahovania. Výsledky jednotlivých prahovacích postupov sú vzápätí zobrazené. Každý z týchto skriptov v sebe obsahuje príkaz na zmeranie jeho dĺžky pre možnosť kontroly časovej náročnosti daného

postupu. Časový údaj nie je možné brať ako absolútne číslo, ale k predstave a porovnaniu náročnosti jednotlivých algoritmov postačuje.

Pre realizáciu boli zvolené nasledujúce druhy prahovania: prahovanie metódou Otsu, prahovanie s hystereziou, prahovanie s využitím informácie o entropii histogramu, lokálne prahovanie s využitím mediánu aj smerodatnej odchylky. Výber konkrétnych metód pre realizáciu bol vedený snahou o porovnanie rozličných prístupov k otázke prahovania s ohľadom na hodnotenie ich výsledkov v práci Sezgina a Sankura [6]

5.1 Prahovanie s využitím mediánového filtra

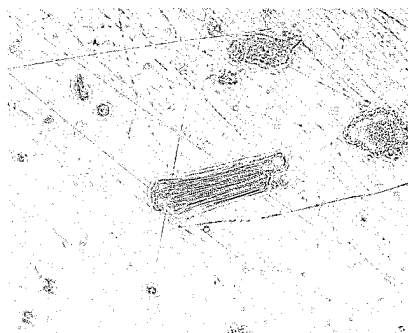
Táto prahovacia metóda patrí medzi lokálne prahovacie metódy, ktorých princíp je popísaný bližšie v kapitole 4.5.3 Funkcia `medfilt2` počíta v blízkom okolí bodu hodnotu mediánu jasovej úrovne a dosadí ju za hodnotu jasu daného bodu. Veľkosť tohoto okolia je možné nastaviť. Pre vyššiu kvalitu filtrácie je lepšie väčšie okno, z hľadiska časovej náročnosti sa snažíme jeho veľkosť minimalizovať. Po filtrácii vypočíta funkcia `med_th` rozdiel filtrovaného a pôvodného obrazu a pomocou funkcie `imcomplement` vypočíta jeho doplnok. Výsledný obraz pri veľkosti okna 5x5 pixelov vidíme na Obr. 5.2(a).

5.2 Prahovanie pomocou priemerovacieho filtra

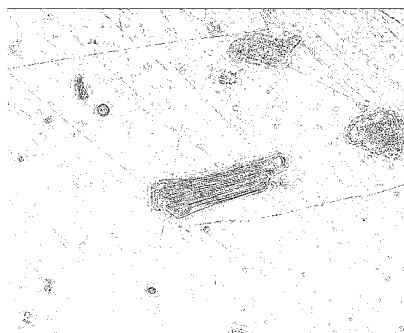
Lokálna prahovacia metóda líšiaca sa od predchádzajúcej len v použitom štatistickom výpočte v rámci filtračného okna. Tu sa namiesto mediánu funkcie počíta aritmetický priemer v okolí každého bodu pomocou funkcie `mean_th`. Nasleduje opäť odčítanie obrazov a vytvorenie doplnku. Obraz vytvorený touto metódou pri veľkosti okna 3x3 vidíme na Obr. 5.2(b).

5.3 Metóda Otsu

V tomto prípade sa jedná o druh globálneho prahovania, kde je pre výpočet prahovej hodnoty využitá metóda Otsu. Jej princíp je bližšie popísaný v kap. 4.5.1. Je jednou z najuniverzálnejších a najpoužívanějších pre výpočet optimálneho prahu. Pre jeho získanie využíva štatistické metódy a výpočet je uskutočnený podľa vzťahu (4.2). Realizácia tejto metódy je v programe Matlab naprogramovaná ako funkcia `graythresh` v rámci *Image processing toolbox*. Výsledný binárny obraz je na Obr. 5.3



(a) mediánový

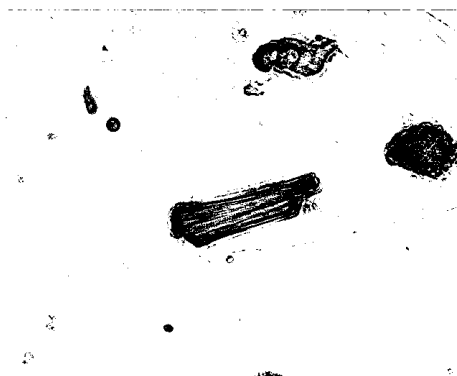


(b) priemerovací

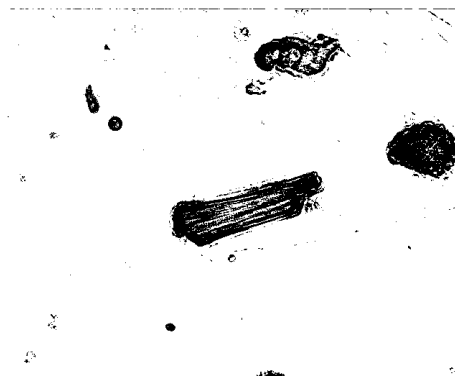
Obr. 5.2: Prahovanie s využitím filtrov

5.4 Prahovanie s hysteréziou

Pre prahovanie s hysteréziou platia pravidlá popísané v kap. 4.3. Pre T_{LOW} je použitá hodnota, ktorá bola vypočítaná metódou Otsu, miera hysterézie δ je určená experimentálne. Vstupnými premennými funkcie `hyster_th` sú hodnota prahu z funkcie `graythresh` a prahovaný obraz. Miera hysterézie sa nastaví vo vnútri tejto funkcie. Pri danom spracovávanom obraze a prahu vypočítanom metódou Otsu mala miera hysterézie len malý vplyv na výsledný binárny obraz. Vďaka tomu, že prah T_{LOW} bol vypočítaný predchádzajúcou metódou, sa výsledné obrazy na seba veľmi podobajú. Výsledok vidíme na Obr. 5.4



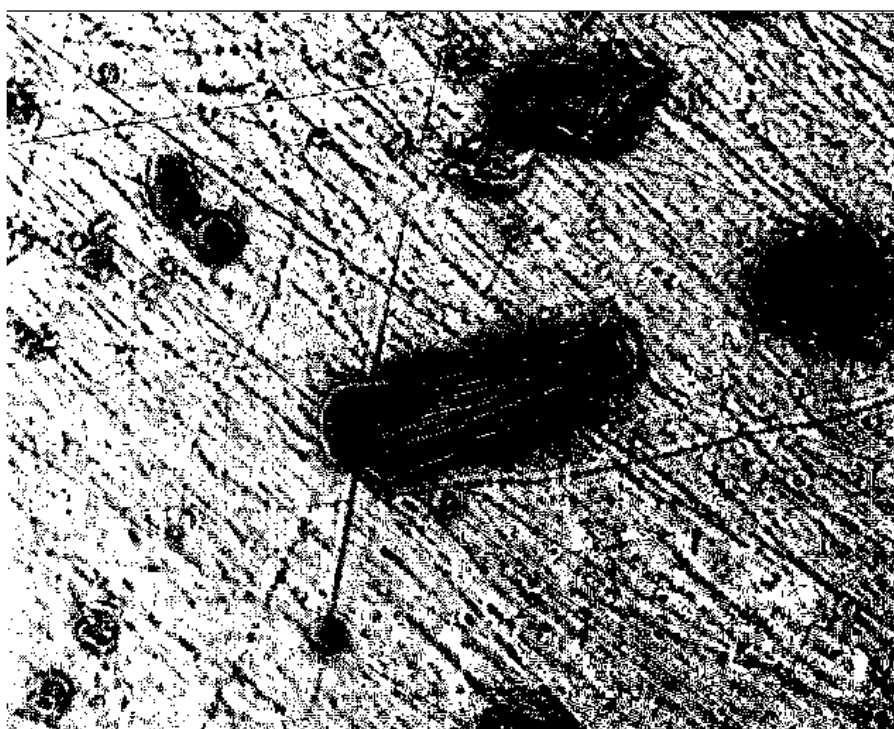
Obr. 5.3: Prahovanie metódou Otsu



Obr. 5.4: Hysterézne prahovanie

5.5 Iteračné prahovanie

Metóda podľa Riddlera [6] patrí do skupiny metód založených na zhlukovaní, ktoré sú popísané v kapitole 4.6.2. Je to pokročilejšia z iteračných metód založených na dvojtriednych Gaussovských modeloch. V n -tom kroku iterácie sa stanoví nová hodnota prahu T_n pomocou aritmetického priemeru hodnôt popredia a pozadia. V praxi je iteračný cyklus ukončený, keď výsledok rozdielu $|T_n - T_{n+1}|$ je dostatočne malý. K výpočtu hodnoty optimálneho prahu vo funkcii `riddler_th` bol použitý vzťah (4.3). Výsledná hodnota optimálneho prahu je použitá pre globálne prahovanie obrazu. Finálny obraz vidíme na 5.5

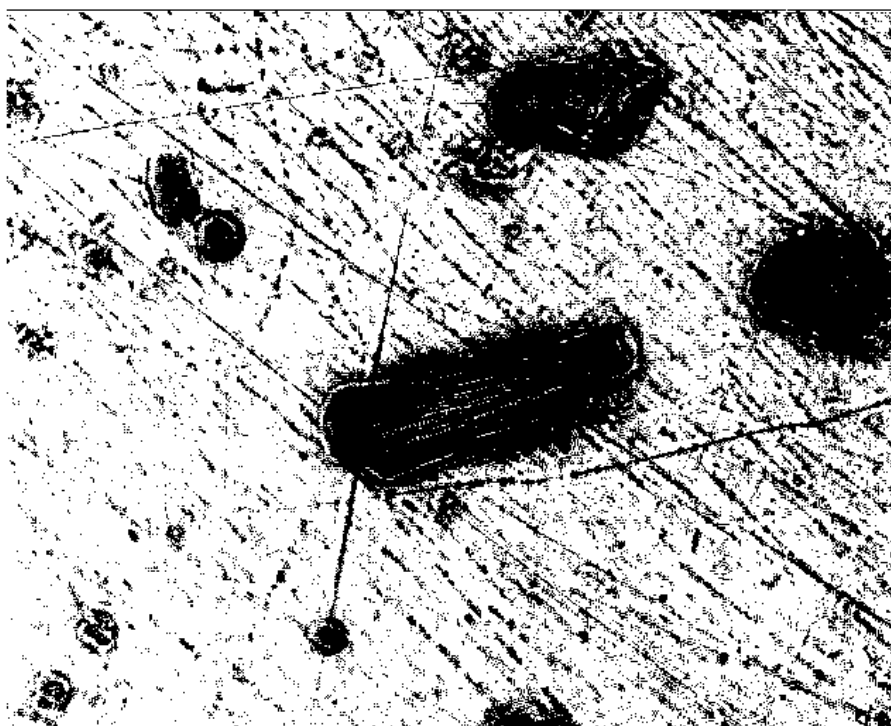


Obr. 5.5: Obraz po iteračnom prahovaní

5.6 Prahovanie s využitím informácie o entropii obrazu

Pri tejto metóde popísanej v kap. 4.6.3 dochádza k rozdeleniu obrazu na objekty a pozadie. Pre obidve kategórie je následne spočítaná entropia. Podľa Kapur, Sahoo

a Wong je určený optimálny prah vtedy, keď súčet entropií pozadia a objektov dosiahne svoje maximum. [6]. K samotnému výpočtu bol použitý vzťah (4.4). Funkcia `adaptive_th` vypočíta hodnoty entropie pozadia a objektov pre každú existujúcu úroveň prahu. Vypočítane hodnoty sčíta a nájde maximálny prvok. Pozícia tohto prvku vo vektore určuje ideálnu hodnotu prahu. Výstupnou premennou tejto funkcie je teda opäť hodnota prahu, ktorá ďalej postupuje ako vstupná premenná do procesu globálneho prahovania podľa vzťahu (4.1). Obráz získaný pomocou tejto metódy vidíme na Obr. 5.6

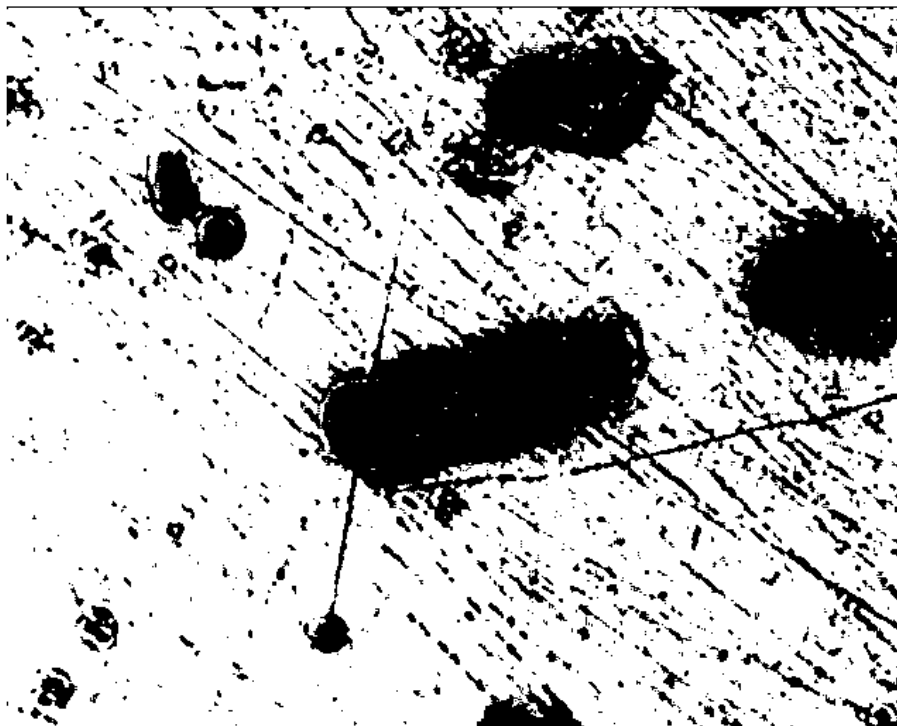


Obr. 5.6: Prahovanie s využitím informácie o entropii obrazu

Mediánová filtrácia

Pre zlepšenie obrazu získaného prahovaním a k potlačeniu šumu v jeho binárnom prevedení môžeme použiť mediánový filter. Tento filter prechádza obraz bod po bode s maskou predvolenej veľkosti. Jeho výstupom je hodnota mediánu z jasových hodnôt bodov nachádzajúcich sa pod maskou. Výsledok filtrácie je teda závislý na veľkosti zvolenej masky a na charaktere konkrétneho obrazu. Zväčšením masky môžeme zlepšiť kvalitu filtrácie, ale dochádza k veľkému zvýšeniu požiadaviek na výpočtový výkon. Experimentálne som zistil, že optimálny výsledok dosahuje funkcia

`medfilt2` pri veľkosti masky 3x3 pixely. Šum v obraze je viditeľne nižší a výpočtové nároky sa pohybujú v akceptovateľných medziach.



Obr. 5.7: Obrázok 5.6 filtrovaný mediánovým filtrom s veľkosťou okna 3x3

6 POROVNANIE VÝSLEDKOV

6.1 Porovnanie výpočtovej náročnosti jednotlivých metód

Všetky metódy, ktoré sú popísané v predchádzajúcich kapitolách som realizoval v programovom prostredí Matlab 7.0.1 pod operačným systémom Windows 7 na prenosnom počítači s procesorom Intel Pentium M s frekvenciou 1,73 GHz s 2 GB operačnej pamäte RAM (PC1). Pre kontrolu nameraných hodnôt som použil rovnakú verziu programu pod operačným systémom Windows 7 64-bit na prenosnom počítači s procesorom Intel Pentium Dual-Core T4300 s frekvenciou 2,1 GHz a operačnou pamäťou s veľkosťou 3 GB (PC2). K odmeraniu času potrebného na výpočet pre každú metódu bol použitý príkaz `tic toc`. Do výsledného času ale nie je zahrnuté predspracovanie obrazu – jeho načítanie a prevedenie do odtieňov šedej, ktorého doba trvania je konštantná. V tabuľke 6.1 sú uvedené hodnoty aritmetického priemeru z desiatich meraní pre každú metódu na oboch počítačoch. Uvedené hodnoty času sú v sekundách. Metódy su očíslované nasledovne:

1. Prahovanie pomocou mediánového filtra 5x5
2. Prahovanie pomocou mean filtra 3x3
3. Prahovanie metódou Otsu
4. Prahovanie s hystereziou
5. Iteračné prahovanie
6. Prahovanie s výpočtom entropie obrazu
7. Binárny obraz z bodu 6. po mediánovej filtrácii

	1	2	3	4	5	6	7
PC1	0,330	0,058	0,060	0,600	0,049	0,027	0,120
PC2	0,232	0,039	0,045	0,390	0,034	0,016	0,085

Tab. 6.1: Časy potrebné na prahovanie

Z tabuľky 6.1 je zrejmé, že pre aplikácie s vysokými nárokmi na rýchlosť spracovania sú výhodné nasledujúce metódy: prahovanie pomocou priemerovacieho filtra, metóda Otsu, iteračné prahovanie podľa Riddlera a metóda prahovania s výpočtom entropie obrazu. Na druhú stranu lokálne prahovanie pomocou mediánového filtra a prahovanie s hystereziou sú so svojimi vysokými výpočtovými nárokmi nevhodné pre použitie v aplikáciách s kontinuálnym zberom obrazových údajov v reálnom čase.

6.2 Subjektívne hodnotenie výsledkov prahovania

Vzniknuté binárne obrazy po aplikovaných prahovacích metódach sa dajú podľa ich charakteru rozdeliť do troch kategórií. Na lokálne prahovaných obrazoch (Obr. 5.2(a) a 5.2(b)) majú objekty v popredí (srdcové bunky) nie príliš výrazné okraje, dokonca aj telá samotných buniek nie sú celé vyplnené čiernou farbou. V prípade použitia priemerovacieho filtra je dobre odfiltrovaný šum pozadia. Výsledok prahovania s použitím mediánového filtra je vzhľadom k jeho veľkej časovej náročnosti úplne nedostatočný.

Druhou skupinou binárnych obrazov sú obrazy vzniknuté prahovaním pomocou metódy Otsu a hysteréznym prahovaním (Obr. 5.3 a 5.4). Výsledné obrazy sa veľmi podobajú, čo je spôsobené tým, že dolná hranica prahu je stanovená pomocou metódy Otsu a nastavoval som iba mieru hysterézie. Po porovnaní výpočtovej náročnosti (stĺpce 3 a 4 z tabuľky 6.1) vidíme, že hysterézia zvyšuje časovú náročnosť zhruba 10-násobne. Na oboch obrazoch sú okraje buniek veľmi ostré, telá sú lepšie vyplnené ako u predchádzajúcich dvoch metód. Šum pozadia je maximálne potlačený. Metóda Otsu teda poskytuje uspokojivý výsledok pri vysokej rýchlosti výpočtu

Iteračné prahovanie a prahovanie s využitím informácie o entropii obrazu poskytujú obrazy so spoločnými znakmi (Obr. 5.5 a 5.6). Objekty popredia – bunky, majú okraje neostré, ale telá sú vyplnená čiernou farbou najlepšie zpočiatku ostatných metód. Pozadie je dosť zašumené, objekty na ňom sa však dajú ľahko identifikovať. Metóda s využitím informácie o entropii obrazu je ale najrýchlejšia zo všetkých realizovaných metód (stĺpec 6 Tabuľka 6.1) a poskytuje priestor na dodatočnú úpravu vzniknutého binárneho obrazu pri relatívne malom zvýšení výpočtovej náročnosti. Na Obr. 5.7 vidíme výsledok aplikácie mediánového filtra s oknom o veľkosti 3x3 na Obr. 5.6. Šum pozadia je potlačený, objekty sú ešte ľahšie identifikovateľné a hrany o niečo ostrejšie. Výsledný čas výpočtu sa predĺži na dvojnásobok času, ktorý potrebuje najpoužívanejšia a najuniverzálnejšia metóda Otsu.

6.3 Objektívne kritériá pre hodnotenie prahovania

Pri automatickom prahovaní obrazu dochádza ku komplikáciám v situáciách, kedy objekty popredia zaberajú nepomerne menšiu oblasť scény ako pozadie alebo keď jednotlivé úrovne jasových intenzít objektov a pozadia majú prekrývajúce sa distribučné funkcie, prípadne je ich histogram unimodálny. [6] V prípade, že je odhad his-

togramu založený na príliš malej vzorke, môže dochádzať k jeho zarušeniu. Následne dochádza k tomu, že jednotlivé body sú zle klasifikované a dochádza k deformáciám tvarov objektov na obraze. Pri prahovaných obrazoch dochádza k zašumeniu objektov aj pozadia. Vzniká tu teda potreba na stanovenie kritérií pre hodnotenie prahovacích algoritmov. Musia brať do úvahy mieru šumu v binárnych obrazoch a tiež zároveň tvarové zmeny objektov.

Medzi kritériá, ktoré nám hodnotia kvalitu prahovacích metód môžeme zaradiť nasledujúce [6]:

Chyba nesprávnej klasifikácie – Misclassification error (ME)

Určuje percento bodov pozadia chybné identifikovanýj ako objekty popredia a naopak, pixelov popredia priradených pozadiu. Pri segmentácii do dvoch tried to môžeme vyjadriť vzťahom:

$$ME = 1 - \frac{|B_O \cap B_T| + |F_O \cap F_T|}{|B_O| + |F_O|} \quad (6.1)$$

kde B_O a F_O znamenajú počty pixelov popredia a pozadia v originálnom obraze a B_T a F_T určujú pixely popredia a pozadia v testovanom obraze. Hodnota ME sa môže meniť v intervale $< 0, 1 >$, kde 0 zodpovedá úplne správne klasifikovanému obrazu a hodnota 1 úplne nesprávne vyhodnotenému obrazu.

Nesprávna detekcia hrany – Edge mismatch (EMM)

Tento spôsob hodnotenia penalizuje nezrovnalosti medzi hranami získanými z pôvodného obrazu a hranami z prahovaného obrazu. Matematicky je táto metrika vyjadrená nasledujúcim vzťahom:

$$EMM = 1 - \frac{CE}{CE + \omega \left(\sum_{k \in \{EO\}} \delta_k + \alpha \sum_{k \in \{ET\}} \delta_k \right)} \quad (6.2)$$

kde CE je počet spoločných pixelov hrán na pôvodnom a prahovanom obraze, EO je počet hranových pixelov, ktoré chýbajú v prahovanom obraze a ET je súbor prevyšujúcich pixelov v prahovanom obraze, ktoré sa v pôvodnom obraze nenachádzajú.

Neuniformita regiónu – region nonuniformity (NU)

Táto hodnotiaca metóda nevyžaduje informáciu o pôvodnom obraze a je definovaná ako:

$$NU = \frac{|F_T| \sigma_f^2}{|F_T + B_T| \sigma^2} \quad (6.3)$$

kde σ^2 vyjadruje smerodatnú odchýlku v celom obraze a σ_f^2 vyjadruje smerodatnú odchýlku popredia. Očakáva sa, že dobre segmentovaný obraz bude mať mieru neuniformity regiónu blízku 0, zatiaľ čo v najhoršom prípade sa bude blížiť hodnota NU k jednej.

Penalizácia deformácie obrazu pomocou Hausdorffovej vzdialenosti – modified Hausdorff distance (MHD)

Hausdorffova vzdialenosť môže byť použitá na posudzovanie tvarovej podobnosti medzi originálnym a prahovaným obrazom. Keď existujú dve konečné množiny bodov z originálneho a prahovaného obrazu, ich Hausdorffova vzdialenosť je definovaná ako:

$$\begin{aligned} H(F_O, F_T) &= \max \{d_H(F_O, F_T), d_H(F_T, F_O)\} \\ \text{kde } d_H(F_O, F_T) &= \max_{f_O \in F_O} d(f_O, F_T) \\ &= \max_{f_O \in F_O} \min_{f_T \in F_T} \|f_O - f_T\| \end{aligned} \quad (6.4)$$

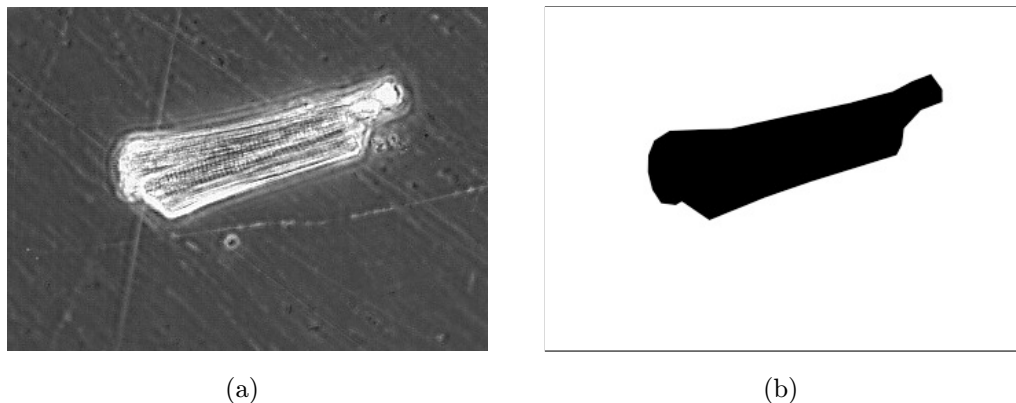
a $\|f_O - f_T\|$ určujú Euklidovskú vzdialenosť dvoch pixelov pôvodného a prahovaného obrazu.

$$MHD(F_O, F_T) = \frac{1}{|F_O|} \sum_{f_O \in F_O} d(f_O, F_T) \quad (6.5)$$

6.4 Vyhodnotenie kvality vybraných prahovacích metód pomocou objektívnych kritérií

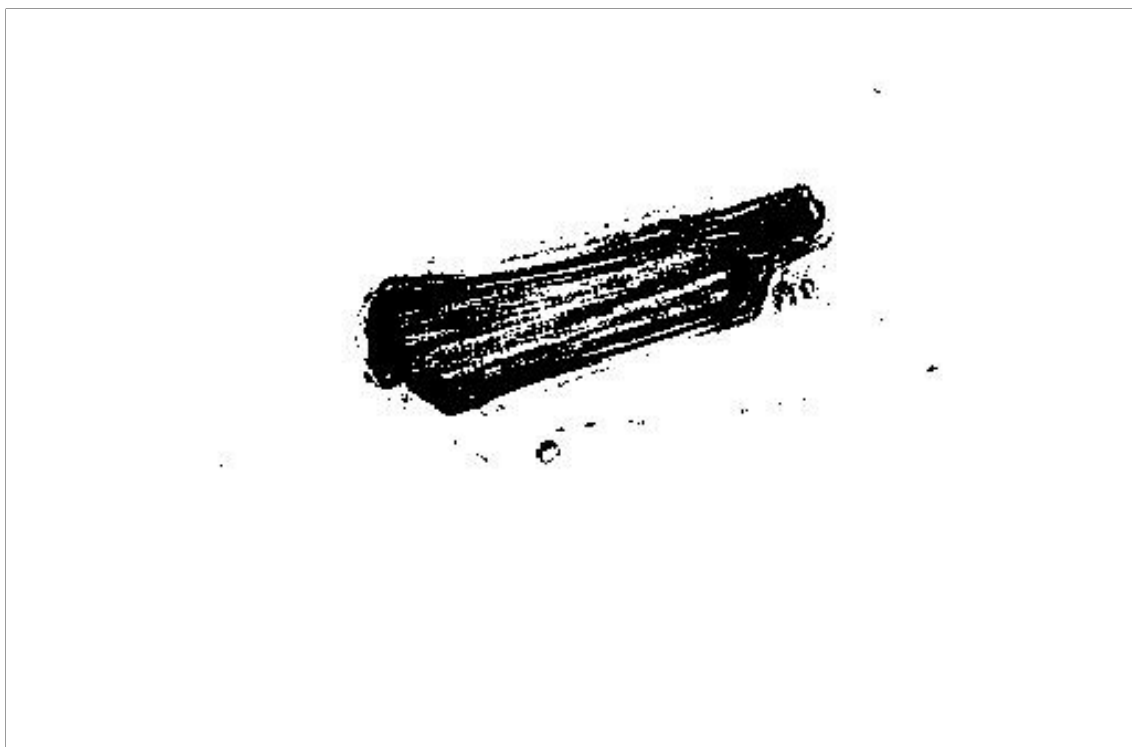
Metódy určené k objektívnemu zhodnoteniu kvality prahovania sú obecné náročné na realizáciu. Niektoré z nich sa ani nedajú aplikovať na prahované snímky – metóda nesprávnej detekcie hrany – pretože pôvodný obraz z optického mikroskopu nie je dostatočne ostrý a nie sú na ňom jednoznačne identifikovateľné hrany. Pre zhodnotenie kvality prahovania pomocou jednotlivých algoritmov som teda zvolil metódu nesprávnej klasifikácie pixelu. Ako už bolo spomenuté vyššie medzi jednotlivými metódami existuje vždy istá miera korelácie, takže získané výsledky by sa mali dostatočne priblížiť výsledkom komplexného hodnotenia pomocou všetkých objektívnych kritérií. Chyba nesprávnej klasifikácie pixelu počíta pomocou vzťahu (6.1) percento nesprávne klasifikovaných pixelov pozadia ako objektov a naopak. Pre hodnotenie a výpočet som použil výrez z pôvodného snímku 5.1 na ktorom je len jeden objekt, ktorý považujeme za objekt popredia, zvyšok je pozadie. Pomocou grafického editora som vytvoril obraz, ktorý zodpovedá ideálne prahovanému snímku 6.1(a). Výsledok vidíme na Obr. 6.1(b)

Počet čiernych pixelov na Obr. 6.1(b) nám udáva počet pixelov, ktoré majú byť identifikované ako súčasť objektu (v našom prípade bunky) ich doplnok v množine

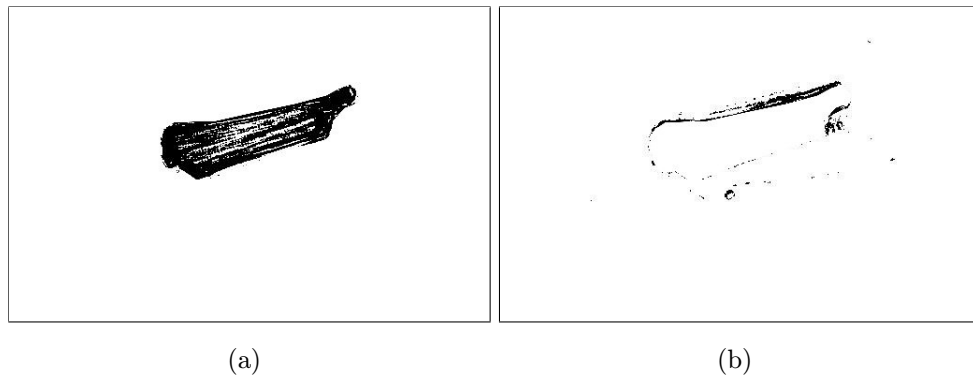


Obr. 6.1: Výrez z Obr. 5.1 (vľavo) a jeho ideálny binárny obraz(vpravo)

350x250 pixelov potom určuje počet bodov pozadia. V grafickom editore som ďalej vytvoril pre každú prahovaciu metódu dva čiastkové obrazy, z ktorých prvý nám pomáha pri výpočte pixelov nesprávne kategorizovaných ako popredie. Z prahovaného obrazu je odstránená oblasť, ktorá v ideálne prahovanom snímku 6.1(b) zodpovedá oblasti bunky. Druhý čiastkový obraz nám zase pomôže pri výpočte pixelov, ktoré sú súčasťou objektu, ale sú nesprávne klasifikované ako pozadie. Tento obraz vznikol po odstránení tej časti obrazu, ktorá na Obr. 6.1(b) neobsahuje čierne pixely. Príklad takýchto dvoch pomocných obrazov vidíme na Obr. 6.3



Obr. 6.2: Výsledok prahovania výrezu z Obr. 6.4 pomocou metódy Otsu



Obr. 6.3: Pomocné obrazy určené k výpočtu hodnoty ME

Údaje o jednotlivých počtoch pixelov z uvedených obrazov nám poslúžia pre výpočet percenta nesprávne klasifikovaných pixelov podľa vzťahu (6.1). V programovom prostredí Matlab som realizoval skript, ktorý počíta hodnoty ME pre každý z uvedených prahovacích algoritmov. Hodnoty ME sa nachádzajú v intervale $\langle 0, 1 \rangle$. Hodnota 0 zodpovedá ideálne prahovanému obrazu a naopak hodnota 1 zodpovedá úplne nesprávne binarizovanému obrazu. [6] Výsledky a hodnoty ME pre všetky realizované algoritmy sú uvedené v tabuľke Tab. 6.2.

Prahovacia metóda	Hodnota ME
Mediánový filter	0.1138
Mean filter	0.1034
Metóda Otsu	0.0426
P. s hystereziou	0.0472
Iteračné prahovanie	0.3487
Max. entropia obrazu	0.1835

Tab. 6.2: Hodnoty ME pre jednotlivé prahovacie metódy

V tabuľke vidíme, že hodnoty kritéria ME sa pohybujú v intervale 4-35%. Znamená to teda, že pri metóde Otsu sa v prahovanom obraze nachádzajú 4% nesprávne klasifikovaných pixelov. Druhá metóda z tohoto hodnotiaceho hľadiska je prahovanie s hystereziou. Najhorší výsledok ME dosiahla metóda iteračného prahovania podľa Riddlera, u ktorej dosiahlo množstvo nesprávne klasifikovaných pixelov úroveň 35%.

7 PRAHOVANIE OBRAZU V LABVIEW

Úvod do LabVIEW

Programovacie a vývojové prostredie LabView(z angl. Laboratory Virtual Instruments Engineering Workbench) čiže „laboratórne pracovisko virtuálnych prístrojov“, je produktom americkej firmy National Instruments, ktorá je priekopníkom a najväčším výrobcom v oblasti virtuálnej inštrumentácie, technickej disciplíny, ktorá zažíva veľký rozvoj v oblasti vývoja, výskumu, školstva a priemyslu. Prostredie LabVIEW je vhodné nielen k programovaniu systémov pre meranie a analýzu signálov, riadenie a vizualizáciu technologických procesov rôznej zložitosti, ale aj k programovaniu zložitých systémov. Hlavným cieľom virtuálnej inštrumentácie je nahradiť dočasne alebo trvalo priestorovo, finančne a veľa krát aj časovo náročné využitie technických prostriedkov virtuálnym riešením s použitím programových prostriedkov a hlavne grafickými a vizuálnymi prostriedkami a sprostredkovať tak užívateľovi maximálnu názornosť. Toto riešenie umožňuje rýchle navrhovanie nových aplikácií aj vykonávanie zmien v konfigurácii, čo je pri realizácii skutočnými nástrojmi za pomoci reálnych súčiastok často veľmi nákladné až nemožné. [7]

V rámci tejto práce som mal implementovať do prostredia LabVIEW prahovacie algoritmy navrhnuté a naprogramované v prostredí Matlab v predchádzajúcej časti. Tieto by mali byť využité pre činnosť detektoru kontrakcie srdcových buniek.

7.1 Implementácia prahovacích algoritmov do LabVIEW

Výhody programovacieho a vývojového prostredia LabVIEW boli hlavným dôvodom aby do neho boli zakomponované prahovacie algoritmy naprogramované a otestované v Matlabe. Je to možnosť vytvoriť samostatný blok určený k prahovaniu pomocou vybranej metódy, ktorý potom následne môže byť jednoducho použitý v ľubovoľnom programe využívajúcom v svojej činnosti prahovanie obrazu.

7.1.1 Možnosti realizácie

Základom prahovacích algoritmov sú matematické operácie s maticami, v ktorých sú uložené informácie o obrazoch. Kvôli veľkým rozmerom týchto matíc, ktoré sa rovnajú rozmerom spracovávaných obrazov v pixeloch (každý pixel je jeden prvok v matici), sú matematické operácie s nimi obtiažne z hľadiska výpočtovej náročnosti a tým pádom aj náročné na čas. V programe Matlab, ktorý je určený pre prácu s maticami – MATrix LABoratory – nám tieto operácie nespôsobovali žiadne

komplikácie avšak pri snahe o naprogramovanie týchto algoritmov v programovom prostredí LabVIEW sa ukázali ako kľúčový problém.

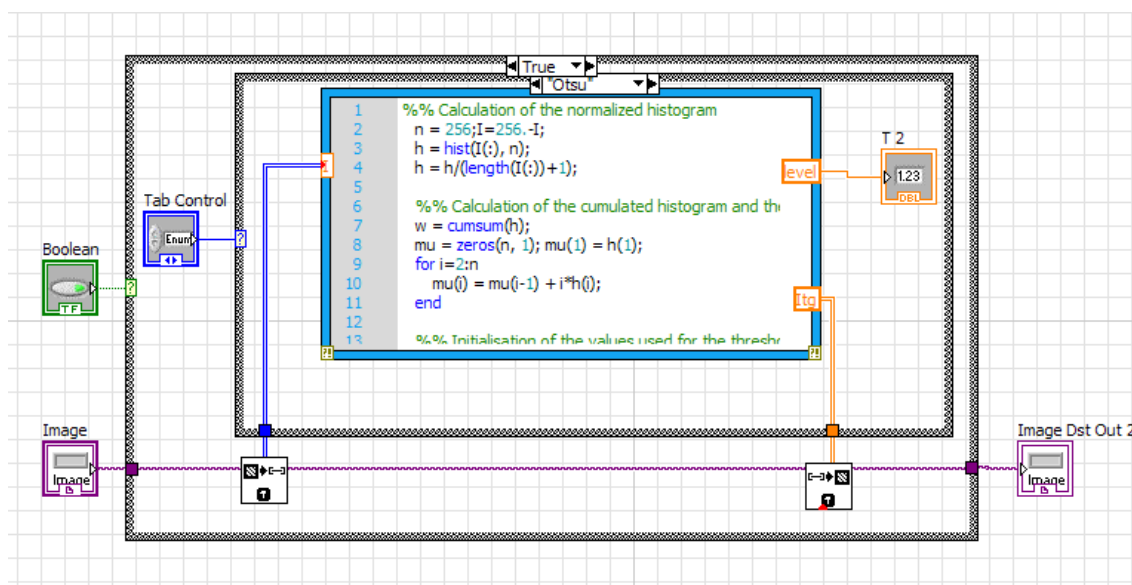
Pre realizáciu prahovacích algoritmov som nakoniec zvolil možnosť využiť **MathScript Node**. V rámci grafického programovacieho prostredia nám ponúka možnosť vkladať textové príkazy, ktoré zodpovedajú príkazom v jazyku Matlab. Zdrojový kód do rámca môže byť zapísaný ručne alebo importovaný z ľubovoľného súboru, napr. M-file. Výpočet prebieha vo vnútri rámca a má ho na starosti nadstavbová súčasť programu s názvom LabVIEW MathScript, ktorý nám umožňuje spúšťať v LabVIEW skripty napísané v jazyku Matlab. Táto alternatíva však nepodporuje všetky príkazy realizovateľné v MathWorks Matlab.

7.1.2 Popis bloku Thresholding.vi

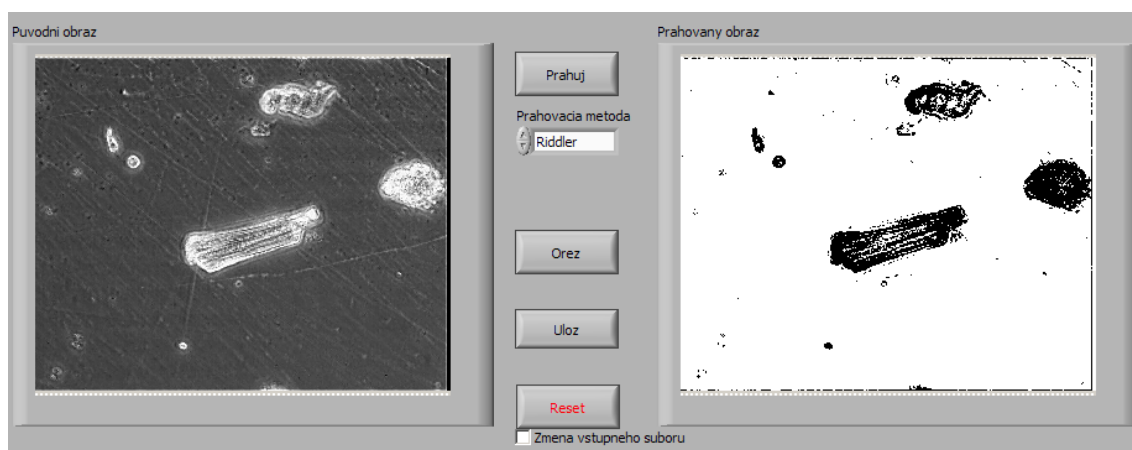
Po prepísaní zdrojových kódov z Matlabu do MathScriptu, ich zjednodušenie a úprave, ktorá bola nutná kvôli vyššie spomenutému obmedzeniu, zostali realizované tri z pôvodných algoritmov. Úspešne bolo realizované iteračné prahovanie podľa Riddlera, prahovanie s využitím informácie o entropii obrazu a prahovanie metódou Otsu. Všetky metódy sú súčasťami štruktúry case a výber konkrétneho typu je realizovaný pomocou prepínača. Každý z použitých **MathScript Node** má jeden vstup a jeden výstup. Vstupom je dvojrozmerné pole, ktoré vytvára z pôvodného obrazu blok **IMAQ ImageToArray**, výstupom je znova dvojrozmerné pole, ktoré je pomocou bloku **IMAQ ArrayToImage** opäť prevedené na obrazový formát. Vstupný a výstupný obraz sú zároveň aj vstupným a výstupným bodom bloku **Thresholding.vi**, ktorými sa tento blok pripája pri použití v ďalších schémach. K vstupom bloku patrí ešte ovládač true/false, ktorý aktivuje alebo deaktivuje samotné prahovanie. Blokovú schému vidíme na Obr. 7.1.

7.1.3 Program prahovanie.vi

Blok popísaný v predchádzajúcej kapitole teraz môžeme použiť napríklad ako súčasť programu pre prahovanie statického obrazu. Tento program bol vytvorený za účelom prahovania obrazov vo formáte jpeg s jednoduchým a užívateľsky príjemným ovládaním. Program obsahuje štyri základné funkčné časti. Prvá z nich sa stará o otvorenie a načítanie súboru, ktorý chceme prahovať. Druhá časť obsahuje bloky, ktoré umožňujú vybrať v obraze oblasť záujmu (ROI - region of interest). Nasleduje časť programu, ktorá obsahuje blok **Thresholding.vi**, teda blok určený k samotnému prahovaniu obrazu. V poslednom funkčnom bloku sa nachádzajú nástroje slúžiace k uloženiu binárneho obrazu. Popis ovládania a galéria obrázkov je uvedená v prílohe A.



Obr. 7.1: Bloková schéma realizácie prahovacích algoritmov



Obr. 7.2: Okno programu pre prahovanie statického obrazu

7.2 Detektor kontrakcie srdcových buniek

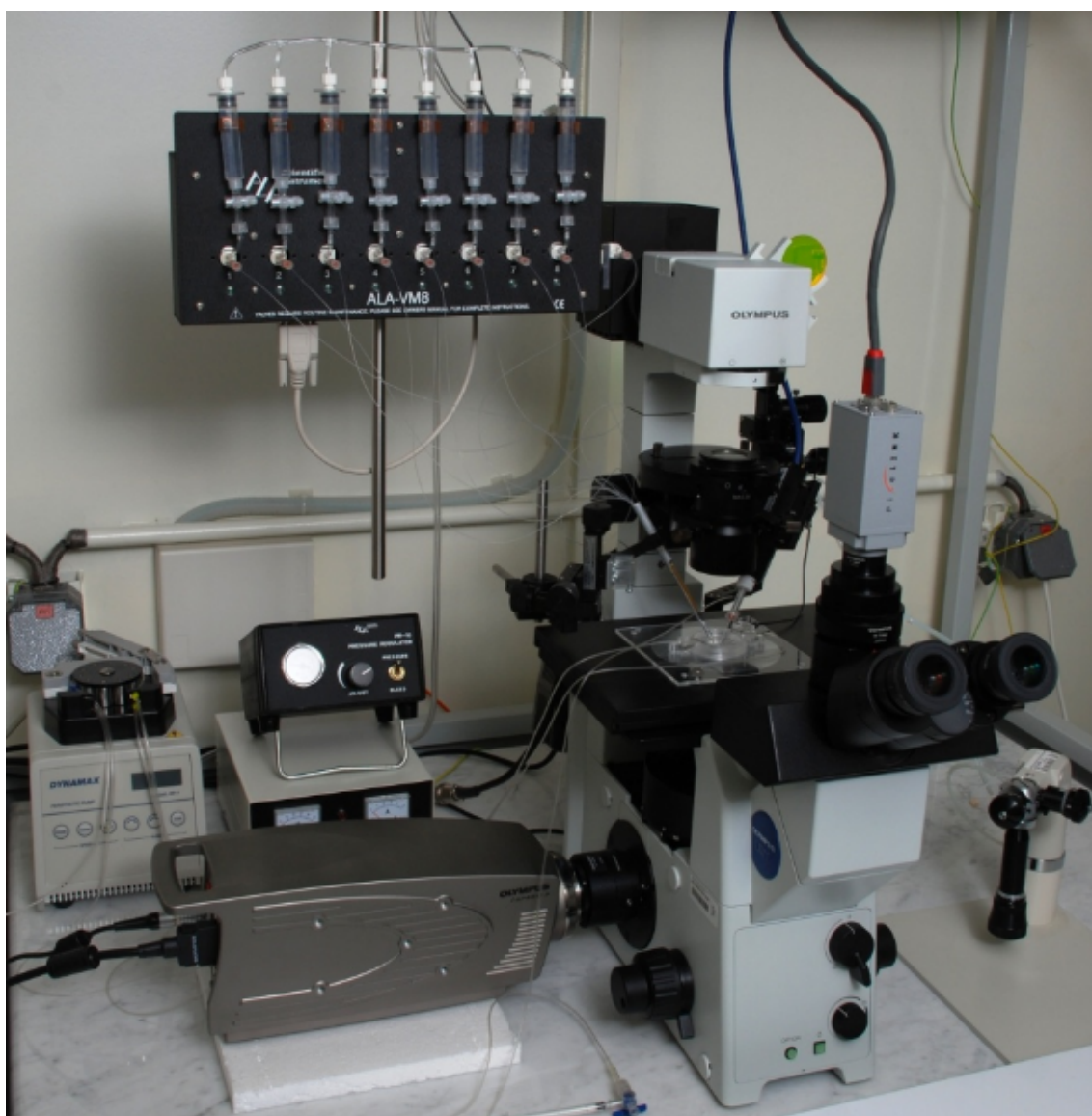
Detektor je prostriedok, ktorý má za cieľ slúžiť ako prostriedok pre riešenie výskumných projektov zameraných na vyhodnocovanie zmeny kontrakcií srdcových buniek v závislosti na dĺžke ich kultivácie, meraním kontraktibility neonatálnych srdcových buniek či obecné vo výskume zaoberajúcom sa problematikou „excitácia-kontrakcia“. [8] Príkladom využitia detektoru je štúdia [9], ktorá sa zaoberá vplyvom látky doxorubicin na mechanizmus excitácie a kontrakcie srdcových buniek morčafa. Doxorubicin je látka obsiahnutá v liekoch proti rakovine a v uvedenej práci autori preukázali negatívny vplyv jej hlavného metabolitu doxorubicinolu. Táto látka vplyva na iontové kanály srdcových buniek a spôsobuje skrátenie trvania akčného

potenciálu a zníženie kontraktibility týchto buniek. Možnú realizáciu meracieho pracoviska pre podobné experimenty potom môžeme vidieť na obrázku 7.3.

Celý systém určený k meraniu by mal obsahovať:

- počítač
- kameru
- mikroskop
- bunkovú lázeň s elektródovým systémom
- software pre detekciu (detektor)
- interný alebo externý prúdový stimulátor

Návrh takéhoto systému je na obrázku 7.4



Obr. 7.3: Realizácia pracoviska pre detekciu kontrakcie srdcových buniek



Obr. 7.4: Návrh systému pre detekciu kardiomyocytov

7.2.1 Realizácia detektoru

Ako základ pre detektor mi bola poskytnutá realizácia od Ing. Vratislava Čmiela, ktorá je popísaná v článku [8]. Pôvodným zámerom bolo nahradiť v ňom originálne prahovacie bloky, za novovytvorené, popísané v kapitole 7.1.2. Avšak už pri realizácii programu pre prahovanie statických obrázkov bola spozorovaná dlhá doba spracovania tohoto bloku. Po odmeraní časov potrebných na beh algoritmov bolo zistené približne 10-násobné predĺženie výpočtových časov pre všetky algoritmy v prostredí LabVIEW v porovnaní s časmi dosahovanými v Matlabe. Príčina tohto problému bola načrtnutá už v kapitole 7.1.1 a jedná sa najmä o tieto dva faktory: Matlab je program určený pre prácu s maticami, jeho výpočty dosahujú maximálnu možnú rýchlosť, takže pri použití iného programovacieho prostredia, bude rýchlosť výpočtu maticových operácií nižšia. Druhým problémom vedúcim k zníženiu rýchlosti bolo využitie funkcie `MathScript Node`, ktorá síce prináša do prostredia LabVIEW možnosti a výhody maticového počtu, ale zase vnáša do celého procesu oneskorenie, ktoré vzniká pri výmene vstupných a výstupných premenných medzi dvoma výpočtovými prostrediami. Pre nemožnosť dosiahnuť rýchlosti spracovania, ktorá by umožnila použitie detektoru v režime realtime, bol teda realizovaný blok z detektoru odstránený a detektor aj naďalej využíva bloky štandardne obsiahnuté v príslušnom module pre prácu s obrazom.

Následne bola venovaná pozornosť úprave tých častí detektoru, ktoré viedli k zlepšeniu funkcií spracovania obrazu a najmä funkcií na vyhodnotenie zmien v snímanom obraze. Popis základných funkčných častí detektoru:

1. Vstupné funkcie

- Otvorenie a načítanie súboru vo formáte avi, prípadne získavanie obrazu z kamery pripojenej k PC
- Rozdelenie videa na jednotlivé snímky

2. Spracovanie obrazu

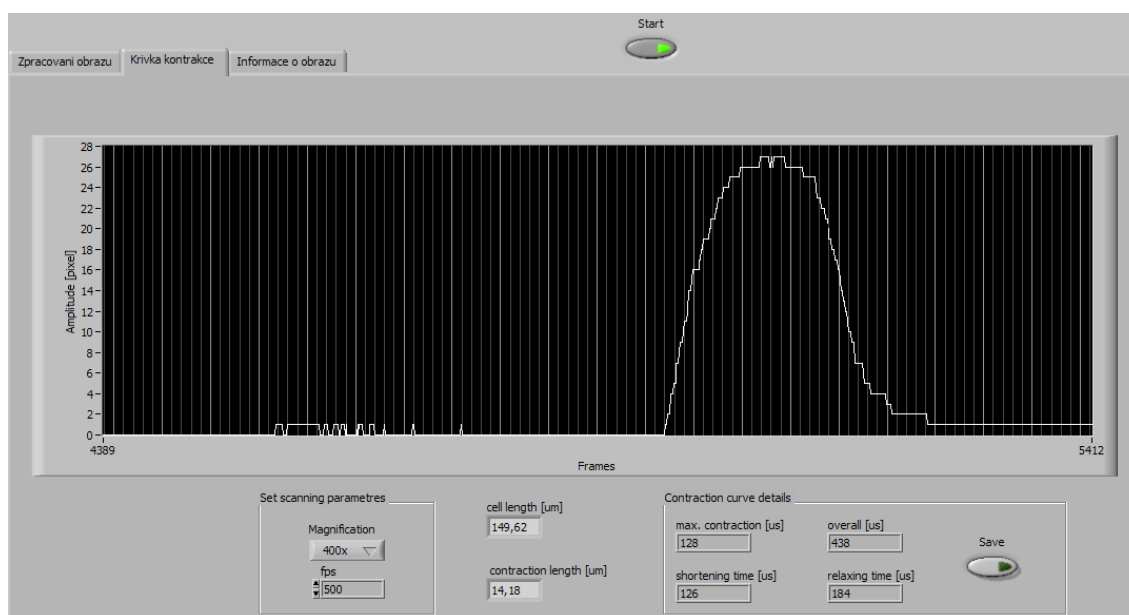
- Výber oblasti záujmu (ROI) - výber časti obrazu, kde sa nachádza bunka, ktorej aktivitu chceme hodnotiť
- Prahovanie - manuálne, automatické, lokálne automatické
- Morfológické operácie - filtrovanie, odstránenie objektov dotýkajúcich sa okrajov, aproximácia objektu pomocou mnohouholníka
- Rotácia - obraz je otočený tak, aby os v ktorej prichádza ku skráteniu objektu bola vo zvislom smere

3. Vyhodnotenie spracovaného obrazu

- Vyhodnotenie kontrakcie - vyhodnotenie výšky objektu a zaznamenanie hodnoty do grafu v závislosti na čase

4. Analýza krivky kontrakcie

- Výpočet parametrov krivky kontrakcie a ich prepočet do skutočných jednotiek podľa nastavených parametrov použitých pri snímaní analyzovaného videa.



Obr. 7.5: Okno programu zobrazujúce krivku kontrakcie

7.2.2 Analýza krivky kontrakcie

Dôležitým grafickým výstupom detektoru je krivka kontrakcie. Vidíme ju na Obr. 7.5. Je to závislosť dĺžky kardiomyocytu na čase. Jej analýzou a porovnaním jednotlivých priebehov môžeme určiť zmeny v aktivite srdcových buniek v závislosti na rozličných faktoroch, ktoré na ňu vplyvajú. Skúma sa vplyv na dĺžku a rýchlosť otvorenia a zatvorenia jednotlivých iontových kanálov v membráne bunky, ktoré významne ovplyvňujú vznik membránových prúdov a tým pádom aj celý mechanizmus kontrakcie. Výstupom programu na záložke **Krivka kontrakcie** je graf, kde je vykreslená závislosť dĺžky bunky v pixeloch na poradí snímok. Nulová úroveň zodpovedá dĺžke úplne relaxovanej bunky, do grafu sa vynáša veľkosť rozdielu aktuálnej a maximálnej dĺžky. Mierka osy y je nastavovaná programom automaticky. Na ose x je zobrazených posledných 1000 hodnôt, ktoré sa postupne posúvajú doľava. Po zadaní údajov o zväčšení mikroskopu a rýchlosti snímania viď Obr. B.4, ktoré boli nastavené pri snímaní videa, program prepočíta hodnoty na reálne rozmery v μm a čas v μs . Pre výpočet skutočných rozmerov bunky bolo nutné experimentálne zistiť rozmer priemetu jedného pixelu snímača do pozorovanej roviny. V prípade použitej kamery Olympus iSpeed 3, kde je rozmer jedného bodu na snímači CMOS $21\mu m \times 21\mu m$ je výpočet skutočnej dĺžky d v μm rovný:

$$d = \frac{n \cdot 21\mu m}{Z} \cdot 10 \quad (7.1)$$

kde n je počet pixelov v obraze a Z je hodnota zväčšenia uvedená na použitom objektíve. Násobenie konštantou 10 reprezentuje nevyužitie 10-násobného zväčšenia okuláru pri snímaní obrazu pomocou pripojenej kamery. V ľavej časti obrázku B.4 vidíme skutočnú dĺžku bunky a absolútnu hodnotu jej zmeny pri poslednej kontrakcii. Údaj o skutočnej dĺžke trvania je vypočítaný ako

$$t = \frac{\text{počet snímok} [-]}{\text{rýchlosť snímania} [s^{-1}]}$$

Po ukončení jedného cyklu kontrakcia-relaxácia sa v spodnej časti obrazovky zobrazia prepočítané parametre krivky, ako sú dĺžka trvania celého cyklu, dĺžka kontrakcie bunky, doba kedy nedochádzalo k ďalšiemu skracovaniu ani uvoľňovaniu a dĺžka doby relaxácie na 90% pôvodnej hodnoty. Tieto parametre vidíme na Obr. B.5

8 ZÁVER

Cieľom mojej práce bolo realizovať detektor kontrakcie srdcových buniek v programe LabView. V úvode práce som sa oboznámil s možnosťami spracovania digitálneho obrazu, bližšie som sa venoval problematike jeho prahovania a detekcie hrán. S nadobudnutými poznatkami som zahájil praktickú časť práce, kde som najskôr realizoval vybrané prahovacie algoritmy v programovom prostredí Matlab. Úspešne sa mi podarilo realizovať šesť rôznych prahovacích metód, ktoré reprezentovali viaceré odlišné prístupy k problematike prahovania. Metódy som vyhodnotil z hľadiska ich časovej náročnosti a kvality prahovania, pričom ako testovací obraz mi slúžili snímky kardiomyocytov z optického mikroskopu. V ďalšej fáze som algoritmy, ktoré dosahovali najlepšie výsledky realizoval v prostredí LabView. Naprogramoval som ich ako samostatný blok a ich funkciu a využil v programe na prahovanie statického obrazu. V tejto fáze som zistil, že rýchlosť spracovania týchto algoritmov už nedosahuje ich rýchlosti v Matlabe a je nedostatočná pre použitie v aplikáciách vyžadujúcich spracovanie v reálnom čase. Pri práci na poskytnutom detektore, u ktorého je požiadavka na prácu v reálnom čase, som sa rozhodol ponechať pôvodné bloky určené k prahovaniu a sústredil som sa na zlepšenie skúmaného obrazu pred samotnou analýzou a hlavne vykresleniu a analýze krivke kontrakcie, ktorá je najdôležitejším výstupom realizovaného detektoru.

LITERATÚRA

- [1] MASOPUST, J. PRŮŠA, R. *Patobiochemie buňky* [online]. Praha. 2003, 130s. [cit. 17. 3. 2010]. Dostupné z URL: <<http://web.telecom.cz/dotdiag/dokument/patobio/bunka.pdf>>.
- [2] ŠMARDA, J. *Biologie pro psychology a pedagogy* [online]. [s.l.]:[s.n.], 2004. 424s. 2004, [cit. 17. 3. 2010]. Dostupné z URL: <<http://www.portal.cz/scripts/detail.php?id=4294>>. ISBN 80-7178-924-0.
- [3] GENČÚR, M. *Nástroj na zpracování fotografovaného textu* [online]. Brno, 2007. 28s. FIT VUT v Brně. Vedoucí bakalářské práce Ing. Lukáš Grulich [cit. 17. 3. 2010]. Dostupné z URL: <<http://www.fit.vutbr.cz/study/DP/rpfile.php?id=4521>>.
- [4] BLÁZSOVITS, G. *Interaktívna učebnica spracovania obrazu* [online]. Bratislava: Knižničné a edičné centrum FMFI UK 2006. 130s. [cit. 17. 3. 2010]. Dostupné z URL: <<http://dip.sccg.sk/>>. ISBN 80-89186-08-4
- [5] HAŠKOVEC, V., MUDROVÁ, M. *Segmentace obrazu s využitím hysterezního prahování*. VŠCHT v Praze, Ústav počítačové a řídicí techniky
- [6] SEZGIN, M., SANKUR, B. *Survey over image thresholding techniques and quantitative performance evaluation*. Journal of Electronic Imaging 13(1), 146-165(January 2004)
- [7] VLACH, J., HAVLÍČEK, J., VLACH, M. *Začínáme s LabVIEW*. – 1. vyd.. – Praha : BEN - technická literatura, 2008. – 247 s. :. ISBN: 978-80-7300-245-9 (brož.)
- [8] ČMIEL, V., RYCHTÁRIK, M., PROVAZNÍK, I. *Detektor kontrakcí izolovaných srdečních buňek*. ÚBMI FEKT VUT v Brně, 2009.
- [9] Ge-Xin Wang, Yong-Xiao Wang, Xiao-Bo Zhou, KORTH M.) *Effects of doxorubicin on excitation-contraction coupling in guinea pig ventricular myocytes*. Institut für Experimentelle und Klinische Pharmakologie und Toxikologie, Abteilung Pharmakologie für Pharmazeuten, UniÖersitätsklinikum Hamburg-Eppendorf, 2001.

ZOZNAM PRÍLOH

A	Program prahovanie.vi	I
A.1	Návod k obsluhu	I
A.2	Galéria obrázkov	II
B	Návod na obsluhu detektoru	IV
B.1	Ukážky funkcií detektoru	V

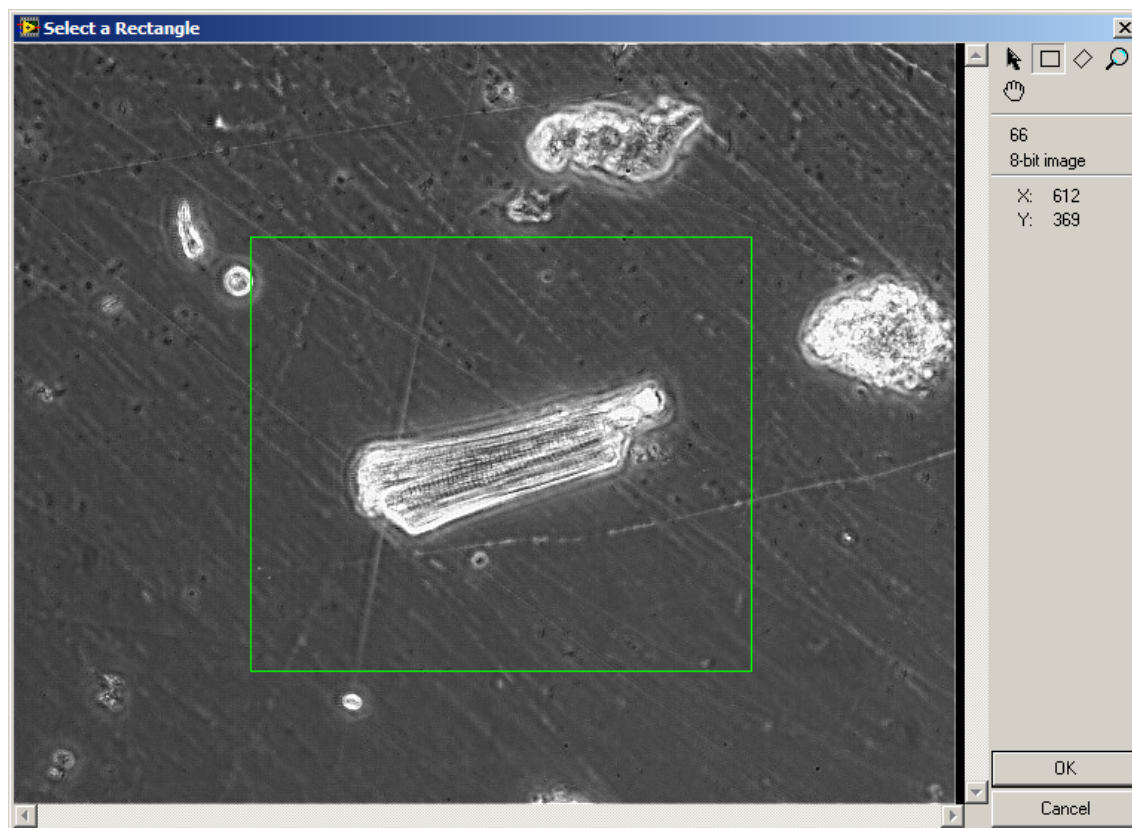
A PROGRAM PRAHOVANIE.VI

A.1 Návod k obsluhu

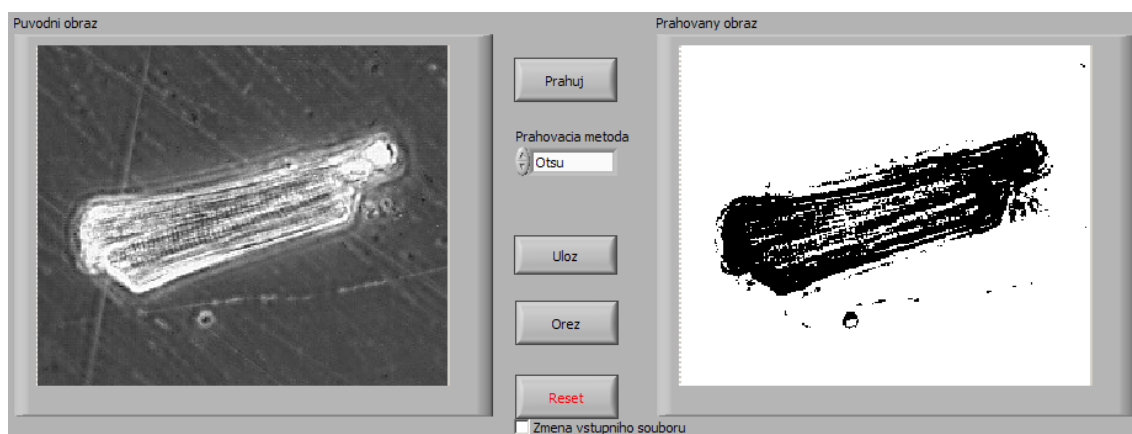
Program prahovanie.vi ponúka možnosť prahovania ľubovoľného obrazu v formáte jpeg v prostredí LabView. Minimálny počet ovládacích prvkov zaručuje jednoduché a intuitívne ovládanie. Po spustení nás program pomocou dialógového okna vyzve k výberu súboru s obrázkom, ktorých chceme prahovať. Následne vidíme základnú obrazovku programu, vľavo sa nachádza okno s originálnym obrazom, v pravej časti sa vyobrazuje výsledný obraz. Úvodná obrazovka je na Obr. 7.2. Medzi týmito dvoma oknami sa nachádzajú ovládacie tlačidlá. Po stlačení tlačidla **Prahu** sa na pôvodný obraz aplikuje prahovacia metóda, ktorá je aktívna v menu pod týmto tlačidlom. Na výber máme iteračné prahovanie, prahovanie metódou Otsu a adaptívne prahovanie s informáciou o entropii obrazu. Po uskutočnení výpočtov sa výsledok vyobrazí v pravom okne. Tlačidlo **Orez** slúži k vybraniu oblasti záujmu pred prahovaním alebo pre priblíženie určitej oblasti v už prahovanom obraze. Dialógové okno tejto voľby vidíme na Obr. A.1. Po stlačení tlačidla **Uloz** sa pomocou dialógového okna ponúkne možnosť uložiť obraz z pravého okna na disk. Tlačidlo **Reset** slúži k odstráneniu všetkých uskutočnených zmien a pri súčasne aktívnom políčku **Zmena vstupního souboru** po jeho stlačení máme možnosť vybrať si k spracovaniu iný obrázok.

Program bol vytvorený za účelom prahovania obrazov buniek z optického mikroskopu, ale jeho využitie môže byť aj v iných oblastiach, kde je potrebné oddeliť od seba časti obrazu. Jednou z nich je aj metóda OCR - optické rozpoznávanie skenovaného textu. Príklad takéhoto využitia je na Obr. A.3

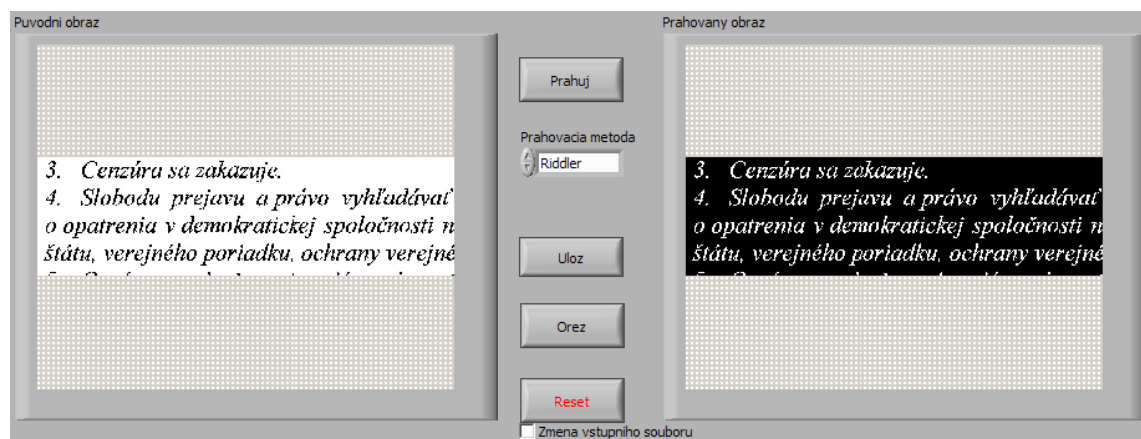
A.2 Galéria obrázkov



Obr. A.1: Dialógové okno pre výber oblasti záujmu



Obr. A.2: Prahovanie vybranej oblasti záujmu



Obr. A.3: Príklad prahovania naskenovaného textu

B NÁVOD NA OBSLUHU DETEKTORU

Po spustení programu nás detektor vyzve dialógovým oknom k výberu oblasti záujmu (ROI) v analyzovanom videu. K neskoršej zmene ROI slúži tlačidlo **ROI selection**, ktoré sa nachádza na prvej karte programu. Táto má názov **Zpracovani obrazu** a nachádzajú sa na nej dve okná pre originálny a prahovaný obraz a nastavenie funkcií spracovania obrazu:

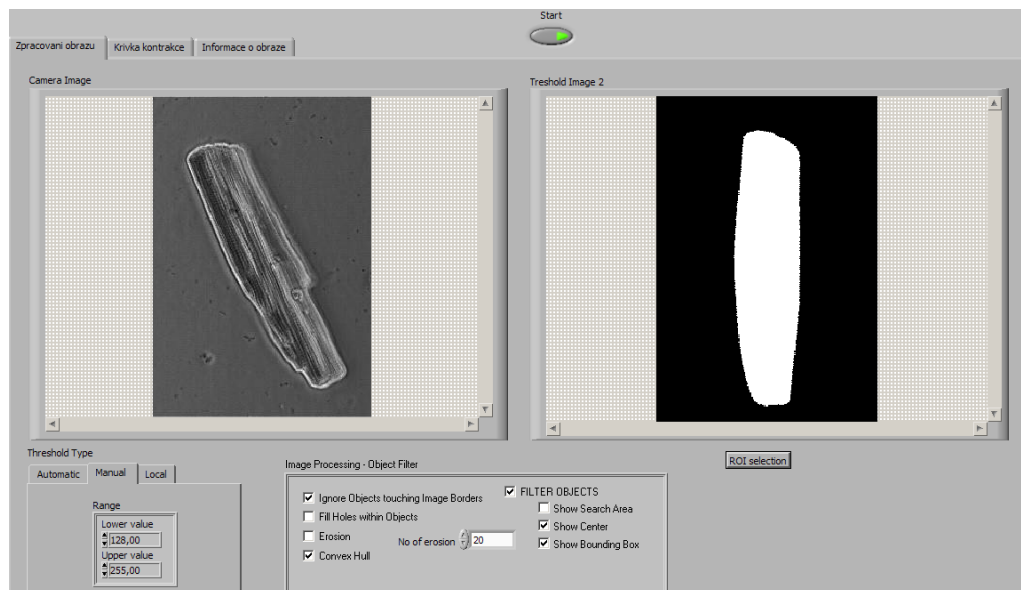
1. Prahovanie
 - manuálne prahovanie podľa histogramu zadáním hornej a dolnej hranice
 - automatické prahovanie
 - automatické lokálne prahovanie
2. Spracovanie pomocou morfologických operácií
 - odstránenie objektov dotýkajúcich sa okrajov obrazu
 - eliminácia malých objektov v obraze - erozie
 - vyplnenie prázdnych miest v objekte
 - aproximácia objektu mnohouholníkom

Na druhej záložke s názvom **Krivka kontrakcie** sa nachádza graf s vykreslovanou krivkou kontrakcie, box s nastavením parametrov pri akých bolo analyzované video nasnímané - zväčšenie mikroskopu a rýchlosť snímania. Na týchto hodnotách sú závislé výpočty parametrov krivky, ktoré sa tiež pod grafom nachádzajú. Po stlačení tlačidla **Save** sa tieto parametre uložia do tabuľky, ktorá sa nachádza na tretej záložke. Spolu s ňou sa tu objavujú ešte dodatočné údaje o obraze, jeho rozlíšenie, histogram a štatistické hodnoty z histogramu. Výsledok vidíme na Obr. B.3

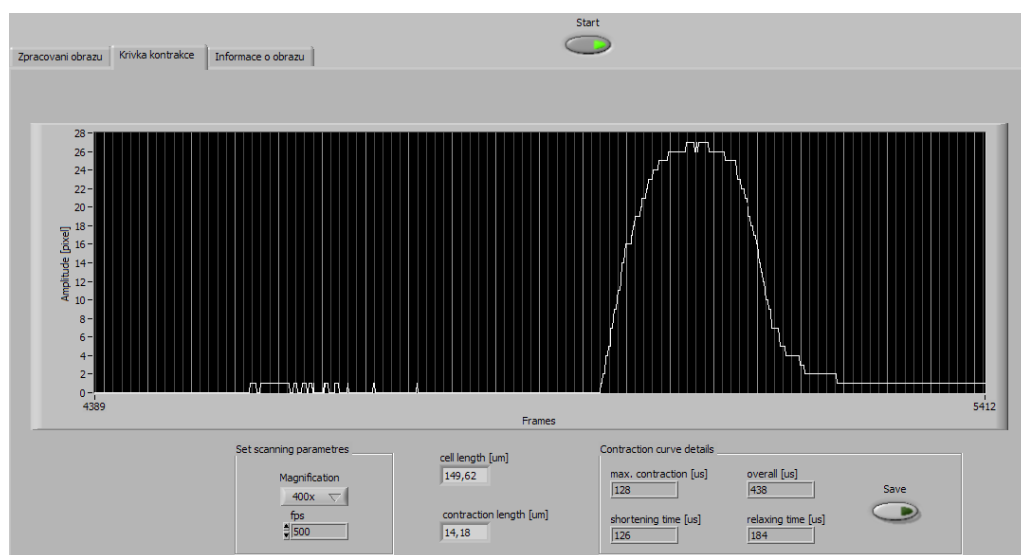
Parametre analyzovaného videa

Video, ktoré spracováva detektor je zaznamenané pomocou vysokorýchlostnej kamery Olympus i-SPEED 3. Kamera ukladá záznam vo formáte hsv – High-Speed Video. Rozlíšenie videa môže byť maximálne 1280 x 1024 bodov v závislosti od nastavenej rýchlosti snímania, ktorá môže byť až 500 obrázkov za sekundu (fps). Parametre nasnímaného videa sú uložené v samostatnom súbore. Pomocou software od výrobcu kamery potom prevedieme video do formátu avi, ktorý je vhodný na ďalšie spracovávanie.

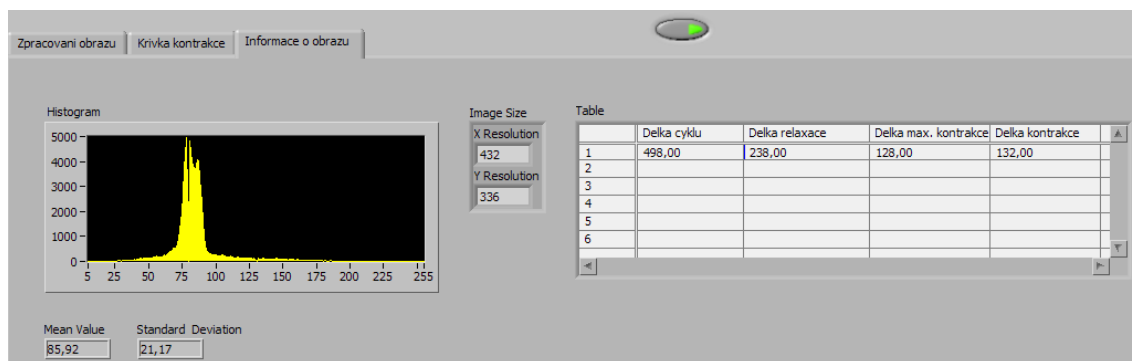
B.1 Ukážky funkcií detektoru



Obr. B.1: Záložka detektoru s možnosťami spracovania obrazu



Obr. B.2: Karta programu s vyobrazenou krivkou kontrakcie



Obr. B.3: Histogram a rozlíšení analyzovaného obrazu

The "Set scanning parameters" dialog box contains the following settings:

- Magnification:** 400x (dropdown menu)
- fps:** 500 (spin box)
- cell length [um]:** 149,62 (text field)
- contraction length [um]:** 14,18 (text field)

Obr. B.4: Detail boxu pre nastavenie snímacích parametrov

The "Contraction curve details" dialog box contains the following settings:

- max. contraction [us]:** 128 (text field)
- overall [us]:** 438 (text field)
- shortening time [us]:** 126 (text field)
- relaxing time [us]:** 184 (text field)
- Save:** A button with a green arrow icon.

Obr. B.5: Detail boxu s prepočítanými dĺžkami jednotlivých častí krivky